

# Diplomarbeit

## **Erprobung Automotive-Radarsensoren für die Bestimmung der Fahrzeuggeschwindigkeit von Schienenfahrzeugen**

eingereicht von

Xin Liang

geb. am 22. August 1990 in Jiangsu

Hochschullehrer: **Prof. Dr.-Ing. O. Michler**

Betreuer: **Dipl.-Ing. Hagen Ußler (TU Dresden)**

**M.Sc. Patrik Havrila (DLR Braunschweig)**

Dresden, den 15.03.2018 \_\_\_\_\_

Matrikel-Nr.: 4017220



## **Aufgabenstellung**

zur Diplomarbeit von Herrn cand. Ing. Xin Liang

Thema:

### **Erprobung Automotive-Radarsensoren für die Bestimmung der Fahrzeuggeschwindigkeit von Schienenfahrzeugen**

Zum Test und zur Entwicklung von Systemen für eine kontinuierliche, schienenfahrzeugseitige Ortung hat das Institut für Verkehrssystemtechnik des Deutschen Zentrums für Luft- und Raumfahrt e.V. (DLR) Braunschweig das Versuchs- und Messfahrzeug RailDrIVE® aufgebaut. Dieses Zweivegefahrzeug ermöglicht als LKW die Vorbereitung von Versuchen, bevor es als Schienenfahrzeug direkt auf der gewünschten Teststrecke eingesetzt wird. Das RailDrIVE® bietet eine flexible, anpassungsfähige Plattform für Versuche zur Ortung von Schienenfahrzeugen und Zustandserfassung des Oberbaus im Schienenverkehr. Dafür sind als Distanz- und Geschwindigkeitsmesser Dopplerradare, ein Achsimpulsgeber, ein optischer Sensor etc. installiert.

Für die Bestimmung der Fahrzeuggeschwindigkeit stehen verschiedene etablierte Sensoren für Schienenfahrzeuge zur Verfügung (Odometer, Doppler-Radare). Die Installation dieser Sensoren erfolgt unter dem Wagenkasten, ist Zulassungsrelevant und die Sensoren sind üblicherweise Teil der sicherheitskritischen Ausrüstung des Fahrzeuges. Die Messdaten können nicht für andere Zwecke abgegriffen werden. Moderne Automotive-Radarsensoren stellen aufgrund ihrer geringen Größe und des Preises sowie der Möglichkeit zur Installation auf dem Wagenkasten eine Möglichkeit zur vorübergehenden Ausrüstung von Schienenfahrzeugen für Messkampagnen dar. Nachteil der Sensoren ist, dass diese üblicherweise in Fahrzeugmitte mit freier Sicht in Fahrtrichtung montiert werden. Im Rahmen dieser Arbeit soll die Eignung von Automotive-Sensoren für den Einsatz auf Schienenfahrzeugen untersucht werden. Insbesondere ist zu untersuchen, ob die Sensoren auch mit Blickrichtung schräg nach vorne und hinten zuverlässig für die Bestimmung der Fahrzeuggeschwindigkeit eingesetzt werden können.

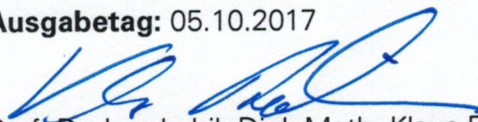
Im Einzelnen sind die folgenden **Aufgaben** zu lösen:

- Literaturrecherche zu Radar- und Geschwindigkeitssensoren für Automotive- und Bahnanwendungen
- Installation der Sensoren auf dem Zwei-Wege-Fahrzeug RailDrIVE®
- Programmierung der Datenerfassung über CAN-Bus mit C++ unter Linux
- Planung und Durchführung von Messfahrten
- Kalibrierung und Abgleich der Messdaten mit den vorhandenen Geschwindigkeitssensoren
- Dokumentation der Ergebnisse

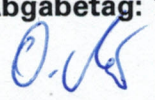
**Betreuer:** Dipl.-Ing. Hagen Ußler (TUD)

M.Sc. Patrik Havrila (Deutsches Zentrum für Luft- und Raumfahrt e.V.)

**Ausgabetag:** 05.10.2017

  
Prof. Dr.-Ing. habil. Dipl.-Math. Klaus Röbenack  
Vorsitzender Prüfungsausschusses

**Abgabetag:** 15.03.2018

  
Prof. Dr.-Ing. Oliver Michler  
Betreuender Hochschullehrer

## **Bibliographischer Nachweis**

Liang, Xin

Erprobung Automotive-Radarsensoren für die Bestimmung der Fahrzeuggeschwindigkeit von Schienenfahrzeugen

Diplomarbeit

Fakultät Elektrotechnik und Informationstechnik und Fakultät Verkehrswissenschaften „Friedrich List“ im Fachbereich Institut für Verkehrstelematik in Zusammenarbeit mit dem Deutschen Zentrum für Luft- und Raumfahrt e.V. Braunschweig.

Professur Informationstechnik für Verkehrssysteme

– 2018 – 60 Seiten – 10 Tabellen, 43 Bilder, 2 Anlage

## **Sperrvermerk**

Diese Arbeit basiert auf vertraulichen, internen Daten des Deutschen Zentrum für Luft- und Raumfahrt e.V. und darf aus diesem Grund ohne Zustimmung des Verfassers und der des Deutschen Zentrum für Luft- und Raumfahrt e.V. nicht von Dritten – mit Ausnahme der wissenschaftlichen Betreuer bzw. der damit beauftragten wissenschaftlichen Mitarbeiter im Prüfungsverfahren, den Prüfern, den Mitgliedern des Prüfungsausschusses sowie den mit der Abwicklung des Prüfungsverfahrens befassten Hochschulbediensteten und im Falle einer rechtlichen Auseinandersetzung dem Justitiariat der TU Dresden, damit befassten Rechtsanwälten, Gutachtern und Gerichten – eingesehen werden.

## Selbständigkeitserklärung

Hiermit erkläre ich, dass ich die vorliegende Diplomarbeit zum Thema:

**„Erprobung Automotive-Radarsensoren für die Bestimmung der Fahrzeuggeschwindigkeit von Schienenfahrzeugen“**

selbständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel genutzt habe. Alle aus fremden Quellen direkt oder indirekt übernommenen Gedanken sind als solche gekennzeichnet.

Dresden, den 15.03.2018

Xin Liang



## **Kurzfassung**

Moderne Automotive-Radarsensoren stellen aufgrund ihrer geringen Größe und des Preises sowie der Möglichkeit zur Installation auf dem Wagenkasten eine Möglichkeit zur vorübergehenden Ausrüstung von Schienenfahrzeugen für Messkampagnen dar. Ein Nachteil der Sensoren ist, dass diese üblicherweise in Fahrzeugmitte mit freier Sicht in Fahrtrichtung montiert werden. Im Rahmen dieser Arbeit soll die Eignung von Automotive-Sensoren für den Einsatz auf Schienenfahrzeugen untersucht werden. Insbesondere ist zu untersuchen, ob die Sensoren auch mit Blickrichtung schräg nach vorne und hinten zuverlässig für die Bestimmung der Fahrzeuggeschwindigkeit eingesetzt werden können.

## **Abstract**

Modern automotive radar sensors, due to their small size and price, as well as the possibility of installation on the vehicle body have provided a possibility for temporary equipment of rail vehicles for measurement campaigns. A disadvantage of the sensors is that they used to be assembled in the vehicle center with clear view in the direction of travel. Through this work the suitability of automotive sensors for use on rail vehicles shall be investigated. In particular, it shall be investigated whether the sensors can also be used reliably for the determination of the vehicle speed, even with a view obliquely forwards and backwards.

## Danksagung

Das Vorhaben wäre nicht möglich gewesen ohne die Unterstützung zahlreicher Einzelpersonen und Institutionen. Mein allerherzlichster Dank gilt dabei Herrn Prof. Dr.-Ing. O. Michler für seine Zusage, der betreuende Professor meiner Diplomarbeit zu werden. Herrn M. Sc. Patrik Havrila danke ich ganz besonders für seine stets großzügige Unterstützung, insbesondere während des Aufenthaltes im DLR. In gleicher Weise möchte ich mich sehr herzlich bei den Mitarbeitern des Institutes für Verkehrssystemtechnik bedanken. Besondere Erwähnung sollen an dieser Stelle auch Herr Dipl.-Ing. Hagen Ußler wegen seiner stets hilfsbereiten Unterstützung und Beratung finden. Für die Korrektur der deutschen Ausformulierung möchte ich hier Joshua Niemeijer nennen.

Meinen lieben Freunden in Deutschland Haitao, Liyu, Yongchao, Joshua, Chrisitian sowie allen anderen nicht namentlich genannten danke ich ganz herzlich für ihre wundervolle Freundschaft und Solidarität, ohne die die Arbeit nicht hätte durchgeführt werden können. In gleicher Weise danke ich allen Freunden in China für ihre moralische Unterstützung. Mein grenzenloser Dank gilt meinen Eltern für ihre besonders wertvolle Unterstützung und ihr Verständnis.

# Inhaltsverzeichnis

Diplomarbeit .....	I
Themenblatt.....	2
Bibliographischer Nachweis .....	3
Sperrvermerk .....	4
Selbstständigkeitserklärung .....	5
Kurzfassung.....	6
Danksagung .....	7
Inhaltsverzeichnis .....	8
Abbildungsverzeichnis .....	10
Tabellenverzeichnis .....	12
Formelzeichen- und Abkürzungsverzeichnis .....	13
1 Einleitung .....	1
2 Grundlagen .....	3
2.1 RailDrivE® .....	3
2.2 Geschwindigkeitssensoren .....	6
2.2.1 Radarsensoren .....	6
2.2.2 Optische Sensoren .....	11
2.2.3 Drehzahlbasierte Sensoren .....	13
2.2.4 Satellitengestützte Sensoren .....	15
2.3 CAN-Bus .....	18
2.3.1 SocketCAN unter Linux .....	19
2.3.2 CANUSB .....	21
3 Vorbereitung und Durchführung von Messfahrten .....	23
3.1 Verbindung des CANUSB.....	23
3.2 Programme zur Speicherung und Dekodierung von CAN-Messages .....	26
3.3 Installation der Sensoren .....	30
3.3.1 Allgemeine Handhabungshinweise der Sensoren .....	30



3.3.2	Installationsanleitung .....	30
3.3.3	Grundbegriffe zur Installationsvarianten.....	32
3.3.4	Installationsvarianten .....	34
3.4	Durchführung der Messfahrten .....	37
4	Algorithmen zur Datenverarbeitung .....	39
4.1	Umrechnung der Radialgeschwindigkeiten .....	39
4.2	Bestimmung der Fahrzeuggeschwindigkeit.....	41
5	Messdatenverarbeitung und Ergebnisdokumentation .....	43
5.1	Sensorblickrichtung schräg nach unten .....	43
5.2	Sensorblickrichtung in Fahrtrichtung .....	46
5.3	Sensorblickrichtung schräg nach vorne und hinten .....	53
6	Zusammenfassung und Ausblick.....	56
	Literaturverzeichnis.....	57
	Anhang .....	61
	Anhang A: Lageskizze des Testgeländes .....	61
	Anhang B: Quellcode des Algorithmus zur Bestimmung der Fahrzeuggeschwindigkeit ....	62

## Abbildungsverzeichnis

Abbildung 2-1: RailDrIVE®: Railway Driving and Validation Environment [3] .....	3
Abbildung 2-2: Anordnung der am RailDrIVE installierten Sensoren [4] .....	4
Abbildung 2-3: Frequenzverschiebung für eine bewegte Quelle [8].....	7
Abbildung 2-4: Geometrische Zusammenhänge zwischen der Lage des Objektes und der Position eines möglichen Sensors [9] .....	8
Abbildung 2-5: Der Bosch Mittelbereichsradarsensor [12].....	9
Abbildung 2-6: Field of View der Antennen [12] .....	10
Abbildung 2-7: Vertikale und Horizontale Version von Luxact (vgl. [13]) .....	11
Abbildung 2-8: Lagerloser Achsimpulsgeber Baumer BPIK [15].....	14
Abbildung 2-9: Javad SIGMA-G3T [18].....	17
Abbildung 2-10: High-Speed-CAN-Bus [21] .....	18
Abbildung 2-11: CANUSB mit dem CAN-Bus verbunden [24] .....	21
Abbildung 2-12: Diagramm der Treibermodelle von CANUSB [25].....	22
Abbildung 3-1: Verbindungsdiagramm .....	23
Abbildung 3-2: Überprüfung des Status der Kernel-Module .....	23
Abbildung 3-3: Liste der Netzwerkschnittstellen .....	25
Abbildung 3-4: Anzeige der CAN-Messages im Terminal.....	25
Abbildung 3-5: Radarkegel der relevanten Bereiche [31] .....	31
Abbildung 3-6: Fahrzeugkoordinatensystem [33] .....	32
Abbildung 3-7: Azimut- und Elevationswinkel der Sensorblickrichtung .....	33
Abbildung 3-8: Azi0Ele0 (Sensorblickrichtung in Fahrtrichtung) .....	34
Abbildung 3-9: Azi45Ele0 (Sensorblickrichtung schräg nach vorne).....	35
Abbildung 3-10: Azi135Ele0 (Sensorblickrichtung schräg nach hinten) .....	35
Abbildung 3-11: Azi0Ele-45 (Sensorblickrichtung schräg nach unten) .....	36
Abbildung 3-12: Route der Messfahrten.....	38
Abbildung 4-1: Radial- und Tangentialkomponenten in Bezug auf Dopplereffekt .....	39
Abbildung 4-2: Umrechnung der relativen Radialgeschwindigkeit .....	40
Abbildung 4-3: Algorithmus zum Rechnen der Fahrzeuggeschwindigkeit .....	41
Abbildung 4-4: IMU-Werte in x-Achse (Fahrzeugfahrtrichtung) .....	42
Abbildung 5-1: Sensorblickrichtung schräg nach unten (vgl. [36]) .....	43
Abbildung 5-2: erkannte Objekte mit Sensorblickrichtung schräg nach unten .....	44
Abbildung 5-3: vergrößerter Teil der Abbildung 5-2.....	45
Abbildung 5-4: erkannte Objekte mit Sensorblickrichtung in Fahrtrichtung.....	46
Abbildung 5-5: vergrößerter Teil der Abb. 5-4 .....	46

Abbildung 5-6: Verteilung von Menge der Objekte mit Sensorblickrichtung in Fahrtrichtung	47
Abbildung 5-7: Geschwindigkeitsverläufe von verschiedenen Sensoren .....	48
Abbildung 5-8: vergrößerter Teil der Geschwindigkeitsverläufe von verschiedenen Sensoren .....	48
Abbildung 5-9: Abweichungen der Geschwindigkeiten zwischen MRR-Sensor und GNSS-Sensor (Javad) .....	49
Abbildung 5-10: Abweichungen der Geschwindigkeiten zwischen MRR-Sensor und Optisch-inertiale-Sensor (Luxact).....	50
Abbildung 5-11: Abweichungen der Geschwindigkeiten zwischen MRR-Sensor und Doppler-Radar (Deuta).....	51
Abbildung 5-12: erkannte Objekte mit Sensorblickrichtung schräg nach vorne .....	53
Abbildung 5-13: vergrößerter Teil der Abbildung 5-12 mit den zwei Situationen.....	54
Abbildung 5-14: Verteilung von Menge der Objekte mit Sensorblickrichtung schräg nach vorne .....	55
Abbildung 5-15: Verteilung von Menge der Objekte mit Sensorblickrichtung schräg nach hinten.....	55
Abbildung A-1: Lageskizze des Testgeländes (vgl. [34]) .....	61

## Tabellenverzeichnis

Tabelle 2.1: Technische Merkmale des MRRs .....	10
Tabelle 2.2: Technische Daten von Luxact [14].....	12
Tabelle 2.3: Leistungsbeschreibungen von Javad SIGMA-G3T [18] .....	17
Tabelle 3.1: Einstellung der CAN-Bitrate [28] .....	24
Tabelle 3.2: Byte-Reihenfolge in Motorola- und Intel-Format [30].....	27
Tabelle 3.3: Signale in der Message mit ID13 .....	28
Tabelle 3.4: Inhalte der CAN-Message mit ID13 .....	29
Tabelle 5.1: Abweichungen der Geschwindigkeiten zwischen MRR-Sensor und Javad .....	49
Tabelle 5.2: Abweichungen der Geschwindigkeiten zwischen MRR-Sensor und Luxact .....	50
Tabelle 5.3: Abweichungen der Geschwindigkeiten zwischen MRR-Sensor und Deuta .....	52

## Formelzeichen- und Abkürzungsverzeichnis

AIM	Anwendungsplattform für Intelligente Mobilität
API	Application Programming Interface
CAN	Controller Area Network
CAN-FD	CAN with Flexible Data-Rate
CORS	Continuously Operating Reference Station
DLL	Dynamic Link Library
DLR	Deutsches Zentrum für Luft- und Raumfahrt
DOP	Dilution of Precision
ETCS	European Train Control System
FTDI	Future Technology Devices International
GNSS	Global Navigation Satellite System
GPS	Global Positioning System
IMU	inertiale Messeinheit, englisch inertial measurement unit
ITVS	Professur für Informationstechnik für Verkehrssysteme
LLCF	Low Level CAN Framework
MRR	mid-range radar
NMEA	National Marine Electronics Association
Radar	radio detection and ranging
RailDrIVE®	Railway Driving and Validation Environment
RFID	Radio-frequency identification
RTK	Real - time kinematic
SIO	Serial Input/Output
SLCAN	Serial CAN Protocol
TDCP	time-differenced carrier phase
VCP	Virtual COM Port
$a_{max}$	maximale absolute Beschleunigung des Fahrzeuges
$\Delta t_{max}$	maximales Zeitintervall zwischen zwei aufeinanderfolgenden Cycles
$v_{Fzg\_Cycle_n}$	berechnete Fahrzeuggeschwindigkeit in dem n-ten Messcycle des MRR-Sensors
$v_{x\_obj}$	umgerechnete Relativgeschwindigkeit des Objektes in Fahrtrichtung des Fahrzeuges

$v_{x\_obj\_benutzbar}$	die benutzbare umgerechnete Relativgeschwindigkeit des Objektes in Fahrtrichtung des Fahrzeuges
$v_{r\_previous}$	die Fahrzeuggeschwindigkeit in dem vorherigen Messcycle des MRR-Sensors
$v_{radial}$	Radialkomponente der Relativgeschwindigkeit
$v_{tangential}$	Tangentialkomponente der Relativgeschwindigkeit

# 1 Einleitung

Moderne Betriebsverfahren wie ETCS erfordern eine fahrzeugbasierte, kontinuierliche Ortung des Fahrzeuges zur Einhaltung von Sicherheitsmerkmalen. Eine falsch detektierte Position eines Fahrzeugs kann zu einer erheblichen Gefährdung führen, da die ermittelte Ortsinformation für die Freigabe und das Wiederbesetzen von Gleisabschnitten genutzt wird. [1] Außerdem ist für regionale Eisenbahnlinien, die normalerweise eine geringe Verkehrsdichte aufweisen, die Ortung on-board der Züge kosteneffektiver als die Bereitstellung von Streckenausrüstung für die Lokalisierung. Eine Kostenreduzierung kann sich ergeben, da nur das On-board-Ortungssystem installiert und gewartet werden muss. Die Wartung der zahlreichen Elemente entlang der Strecke (z. B. Achszähler, Gleisstromkreise) wird nicht mehr benötigt. [2]

Seit ein paar Jahren hat das Institut für Verkehrssysteme des Deutschen Zentrums für Luft- und Raumfahrt (DLR) Methoden und Multisensorsysteme zur Ortung der Schienenfahrzeuge untersucht. Zu diesem Zweck wurde das Zweiwegfahrzeug RailDrIVE® als rollendes Labor zur Erfassung von Sensordaten entwickelt. Basierend auf diesen Daten wurde ein innovativer Algorithmus zur Datenanalyse entwickelt und anschließend unter realen Bedingungen geprüft.

Da die Fahrzeuggeschwindigkeit zur Ortung eines Schienenfahrzeuges eine wichtige Rolle spielt, stehen verschiedene etablierte Sensoren für die Bestimmung der Schienenfahrzeuggeschwindigkeit zur Verfügung (Odometer, Doppler-Radare etc.). Die Installation dieser Sensoren erfolgt unter dem Wagenkasten, ist Zulassungsrelevant und die Sensoren sind üblicherweise Teil der sicherheitskritischen Ausrüstung des Fahrzeuges.

Moderne Automotive-Radarsensoren stellen aufgrund ihrer geringen Größe und des Preises sowie der Möglichkeit zur Installation auf dem Wagenkasten eine Möglichkeit zur vorübergehenden Ausrüstung von Schienenfahrzeugen für Messkampagnen dar. Nachteil der Sensoren ist, dass diese üblicherweise in Fahrzeugmitte mit freier Sicht in Fahrtrichtung montiert werden. Im Rahmen dieser Arbeit soll die Eignung von Automotive-Sensoren für den Einsatz auf Schienenfahrzeugen untersucht werden. Insbesondere ist zu untersuchen, ob die Sensoren auch mit Blickrichtung schräg nach vorne und hinten zuverlässig für die Bestimmung der Fahrzeuggeschwindigkeit eingesetzt werden können.



Die Arbeit ist folgendermaßen aufgebaut:

In Kapitel 2 werden neben der Vorstellung des von DLR entwickelten Versuchs- und Messfahrzeuges RailDrIVE® auch die Grundkenntnisse von verschiedenen Geschwindigkeitssensoren insbesondere für Automotive- und Bahnanwendungen dargelegt. Außerdem werden das für die Datenkommunikation verwendete Protokoll unter Linux und die Hardware-schnittstelle erläutert.

In Kapitel 3 wird erklärt, wie die Messfahrten vorbereitet bzw. durchgeführt wurden. Nachdem die Programme zur Speicherung und Verarbeitung von Messdaten implementiert werden, wird der einzusetzende Automotive-Radarsensor je nach verschiedenen Installationsvarianten montiert. Zuletzt werden die Messfahrten auf dem Testgelände durchgeführt.

In Kapitel 4 werden zunächst die Zusammenhänge zwischen den Eingangsvariablen und der Zielvariable analysiert. Danach werden die entwickelten Algorithmen zur Datenverarbeitung nämlich zur Bestimmung der Fahrzeuggeschwindigkeit vorgestellt.

In Kapitel 5 werden nach der Anwendung der Algorithmen auf die Messdaten die Ergebnisse mit anderen vorhandenen Geschwindigkeitssensoren verglichen, um damit die Messqualität zu bewerten.

In Kapitel 6 gibt es nach der Zusammenfassung einen Ausblick auf mögliche Fortsetzungen der Arbeit.

## 2 Grundlagen

### 2.1 RailDrIVE®

Das Institut für Verkehrssystemtechnik des Deutschen Zentrums für Luft- und Raumfahrt e.V. (DLR) Braunschweig hat das Versuchs- und Messfahrzeug RailDrIVE® für die Erprobung und Entwicklung von Ortungssystemen für Schienenfahrzeuge entwickelt (siehe Abbildung 2-1). Dieses Zweiwegfahrzeug kann als Lastkraftwagen auf die Versuche vorbereitet werden, bevor es als Schienenfahrzeug auf der erforderlichen einzusetzenden Teststrecke gefahren wird. [3]



**Abbildung 2-1: RailDrIVE®: Railway Driving and Validation Environment [3]**

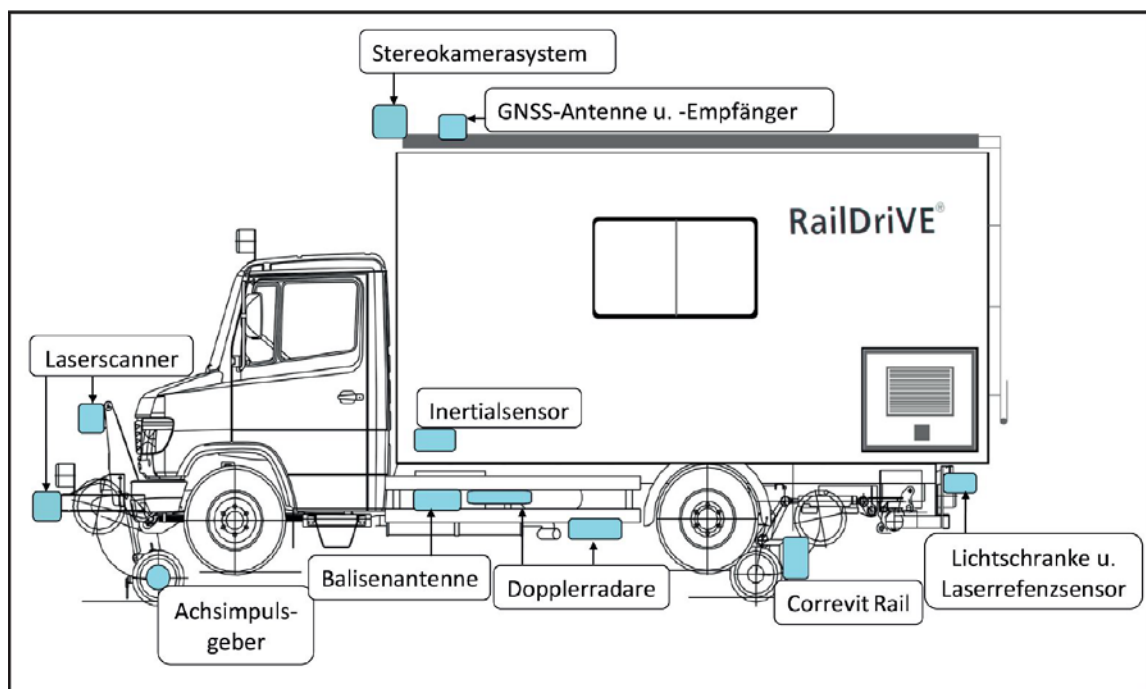
Das RailDrIVE hat vielfältige Anwendungsmöglichkeiten. Durch das Fahrzeug kann die neue Positionierungskomponente in der tatsächlichen Situation getestet werden, um die Auswirkungen von Versagen zu bewerten. Auf Wunsch der Kunden kann das DLR die Entwicklung eines geeigneten Algorithmus zur Datenfusion verschiedener Positionssensoren übernehmen. Außerdem können verschiedene Sensorkombinationen getestet und ausgewertet werden, um die beste Kombination basierend auf Kundenanforderungen zu erhalten.

Das RailDrIVE dient auch als Plattform für die Entwicklung von Anwendungen mit dem zukünftigen europäischen Satellitennavigationssystem Galileo. Zusätzlich zu den Tests und der

Implementierung des Ortungssystems wird das RailDrIVE auch zur Messung der Streckengeometrie verwendet. Darauf basierend können digitale Karten von darauf gefahrenen Bereichen erstellt werden.

Die Kombination aus autonomen Fahrzeugortungs- und drahtlosen Kommunikationssystemen kann verwendet werden, um infrastrukturbezogene Kollisionsvermeidungssysteme zu entwickeln und zu testen. Als zusätzliche Sicherheitssysteme können diese Systeme während des Rangierens oder in den noch nicht technischen abgesicherten Strecken sicherer betrieben werden. Die Erfassung von bestimmten Fehlerzuständen des Gleisoberbaus ist ebenfalls eine Anwendungsmöglichkeit des RailDrIVE.

Weil dieses Zweibegefahrzeug mit einer Reihe von Ortungs- und Kommunikationskomponenten ausgestattet ist, bietet RailDrIVE® eine flexible und anpassungsfähige Plattform für Positionsprüfungen und Statusaufzeichnungen von Gleisoberbauten. Aufgrund der modularen Architektur kann die Basiskonfiguration bei Bedarf in kurzer Zeit erweitert werden.



**Abbildung 2-2: Anordnung der am RailDrIVE installierten Sensoren [4]**

Die Anordnung der am RailDrIVE schon installierten Sensoren wird in Abbildung 2-2 dargestellt. Die Basisausrüstung umfasst eine GNSS-Antenne (Global Navigation Satellite System) und einen Empfänger mit einem GSM-System (Global System for Mobile Communications) sowie ein Funkmodem zum Empfang des Korrektursignals und damit zur Bestimmung der absoluten Position, der Geschwindigkeit und der präzisen Zeit. Zur Messung von Entfernung

und Geschwindigkeit werden Doppler-Radargeräte, Achsimpulsgeber, optische Sensoren und Wirbelstromsensoren installiert. Der Inertialsensor kann Drehrate und die Beschleunigung des Fahrzeugs messen. Eine Balisenantenne, ein RFID-Antennensystem und ein Laserscanner zur Überwachung der Umgebung vervollständigen das fahrzeugseitige Sensorsystem. Zur Grundausstattung gehören außerdem eine separate Stromversorgung und zwei Computerarbeitsplätze mit integrierten Zeitservern. [3]

Mit dem hochwertigen Zeitserver werden ununterbrochene Updates der Zeit erstellt. Alle Komponenten auf und außerhalb dessen nutzen diese Zeit, um ihre Datensätze zu synchronisieren und sie in dem gesamten Protokoll zusammenzuführen. Die Datensätze werden im Rohdatenformat erstellt, um dieselben Datensätze für zukünftige Forschungsthemen zu gewährleisten. Gemäß diesem Gesamtprotokoll kann man Analysen durchführen, um Kausalitäten für den Vorfall zu liefern, die Vor- und Nachteile des Anordnens des Sensors an dem Fahrzeug aufdecken oder vorläufige Hinweise geben, um die Grundlage für die Entwicklung eines neuen Eisenbahnsystems zu legen. [4]

## 2.2 Geschwindigkeitssensoren

Geschwindigkeitssensoren messen die Schnelligkeit eines Körpers und möglicherweise auch die Richtung der Bewegung. Sie lassen sich in zwei hauptsächliche Gruppen unterteilen: Die Doppler-Effekt basierte Geschwindigkeitssensoren. und die Weg-Zeit-Messung basierten Geschwindigkeitssensoren. (Vgl. [5])

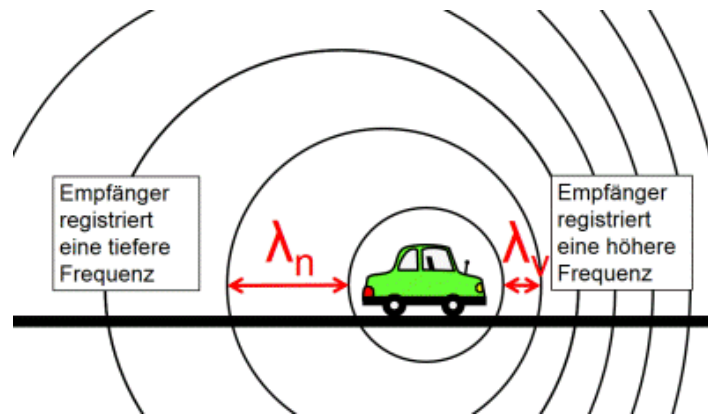
Die folgenden Unterkapitel beschreiben die oben genannten Sensorgruppen. Nach der Erklärung des Messprinzips werden anschließend beispielhaft kommerzielle Geschwindigkeitssensoren mit ihren Spezifikationen aufgeführt.

### 2.2.1 Radarsensoren

#### Dopplereffekt

Radarsensoren werden schon seit vielen Jahren in vielen Anwendungen eingesetzt. Ein bekanntes Anwendungsbeispiel ist eine Radarvorrichtung der Polizei zur Messung bzw. Überwachung der Geschwindigkeit eines Fahrzeugs im Straßenverkehr. Darüber hinaus sind Radarsensoren bereits fester Bestandteil der Flugüberwachung. Sie alle basieren auf dem Grundprinzip des Doppler-Effekts, der 1842 von Christian Doppler entdeckt wurde. Dieses Grundprinzip gilt für alle Arten von Wellen. Daher wurde im Forschungsprozess in diesem Gebiet der Doppler-Effekt von Schallwellen, Wasserwellen, Licht und anderen elektromagnetischen Wellen beobachtet. Wenn sich die Quelle und der Empfänger relativ zueinander bewegen, unterscheiden sich die Frequenz, die von dem Empfänger empfangen wird, und die Frequenz der Welle, die von der Quelle emittiert wird. (vgl. [6], [7]).

Wenn die vom Sender emittierten Wellenfronten auf ein relativ bewegtes Objekt trifft, erfahren die Wellenfront des Signals eine von der Richtung der Bewegung abhängige "Kompression" oder "Ausdünnung", die als eine Frequenzverschiebung auftritt. Dieser Effekt wird in Abbildung 2-3 anschaulich dargestellt.



**Abbildung 2-3: Frequenzverschiebung für eine bewegte Quelle [8]**

Es spielt bei der Betrachtung von elektromagnetischen Wellen keine Rolle, ob sich Sender und/oder Empfänger bewegen. Die resultierende Frequenzverschiebung hängt nur von der relativen Geschwindigkeit  $v$  zwischen dem Sender und dem Objekt ab. Die tatsächliche Verallgemeinerung der elektromagnetischen Wellen beruht auf den Kenntnissen der speziellen Relativitätstheorie, die Einstein unter der Annahme entwickelt hat, dass für die Übertragung der elektromagnetischen Wellen kein Übertragungsmedium nötig ist. Für den relativistischen Doppler-Effekt gilt entsprechend für ein Objekt, das sich vom Sender entfernt, der folgende Zusammenhang [9]

$$f_e = f_0 \sqrt{\frac{c-v}{c+v}} \quad (2.1)$$

Dabei ist  $f_0$  die Frequenz des Senders und  $f_e$  die am Objekt gemessene Frequenz. Weil die Relativgeschwindigkeit  $v$  der Objekte im Vergleich zur Lichtgeschwindigkeit  $c$  ist ( $v \ll c$ ), kann eine Näherung angesetzt werden [10].

$$f_e \approx f_0 \left(1 - \frac{v}{c}\right) \quad (2.2)$$

Nachdem die Welle am Objekt reflektiert wird, kehrt die Welle zum Radargerät zurück und erfährt daher die gleiche Frequenzverschiebung auf dem Rückweg, da in diesem Fall das Objekt selbst als die Sendequelle der Welle betrachtet werden kann. Die absolute Frequenzänderung  $f_D$  zwischen empfangener und ausgesendeter Frequenz beträgt dann

$$f_D = 2 \frac{f_0}{c} \cdot v = 2 \cdot \frac{v}{\lambda} \quad (2.3)$$

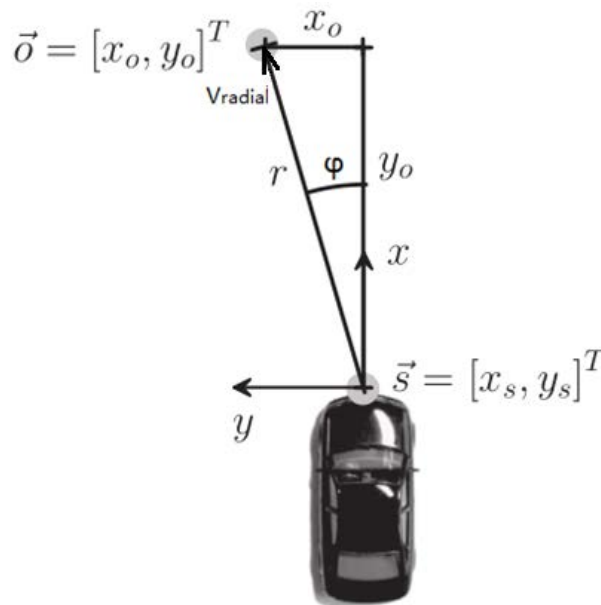
Die Richtung der Geschwindigkeit  $v$  wird gemäß dem verwendeten Koordinatensystem definiert (vgl. [11]), sodass das relativ entfernende Objekt einen positiven Geschwindigkeitswert hat. Bei sich relativ entfernenden Objekten wird eine Verringerung der Frequenz festgestellt.

Eine andere wichtige Größe, die ein Radarsensor bestimmen kann, ist der Abstand  $r$  zwischen Sensor und Objekt. Da der Zielwinkel  $\varphi$ , in dem sich das Objekt befindet, üblicherweise auch gemessen wird, wird die Angabe der mit dem Radar gemessenen Position üblicherweise in Polarkoordinaten ausgedrückt. Angenommen, dass die Abmessung des Objekts klein ist, so ist es sehr einfach die Position des Objekts im kartesischen oder polaren Koordinatensystem durch den Ortsvektor  $\vec{o}$  zu bestimmen. Entsprechend werden die über Ortsvektoren beschriebenen Objekte auch als Punktziel bezeichnet (vgl. Abbildung. 2-4). [9]

$$\vec{o} = \begin{bmatrix} x_o \\ y_o \end{bmatrix} \hat{=} \begin{bmatrix} r_o \\ \varphi \end{bmatrix} \text{ mit } x_o = r_o \cdot \cos(\varphi) \text{ und } y_o = r_o \cdot \sin(\varphi) \quad (2.4)$$

Wenn der Sensor sich an der Position  $\vec{s} = [x_s, y_s]^T$  befindet, so kann die Position des detektierten Objektes auch relativ zu diesem Punkt dargestellt werden. Der relative Abstand zwischen Sensor und Objekt beträgt dann

$$r = |\vec{o} - \vec{s}| = \sqrt{(x_o - x_s)^2 + (y_o - y_s)^2} \quad (2.5)$$



**Abbildung 2-4: Geometrische Zusammenhänge zwischen der Lage des Objektes und der Position eines möglichen Sensors [9]**

Trotz verschiedenen Verfahren zur Entfernungsmessung ist die Grundlage dazu immer die Laufzeit des Signals zwischen Sendung und Empfang. Die Entfernung  $r$  kann auf Basis der Laufzeit  $\tau$  und der Lichtgeschwindigkeit  $c$  somit berechnet werden.

$$r = \frac{c \cdot \tau}{2} \quad (2.6)$$



### **Bosch Mittelbereichsradarsensor (MRR) [12]**

Ein Unternehmen, das sich auf Radarsensoren im Automotive-Bereich spezialisiert hat, ist die Robert Bosch GmbH. Die Hauptaufgabe des Bosch-MRRs ist das Erkennen von Objekten sowie die Messung von deren Geschwindigkeit und Position in Relation zur Bewegung des eigenen Fahrzeugs. Durch den Mittelbereichsradarsensor (MRR) können Fahrzeughersteller eine Reihe von Komfort- und Sicherheitsfunktionen in ihren Fahrzeugen implementieren. Abbildung 2-5 zeigt das Aussehen des Produktes.

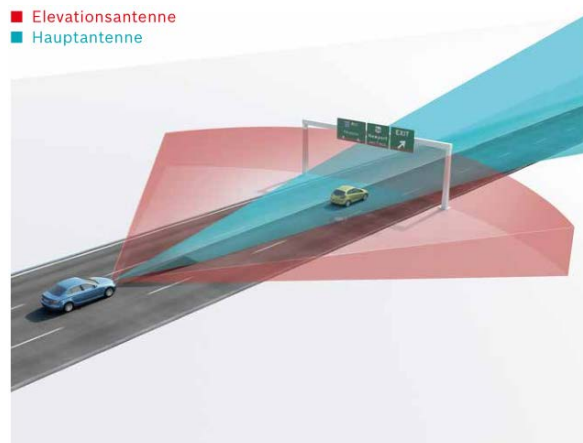


**Abbildung 2-5: Der Bosch Mittelbereichsradarsensor [12]**

Der Bosch MRR nutzt einen globalen Frequenzbereich von 76-77 GHz für Radaranwendungen in der Automobilindustrie. Im Vergleich zu den bestehenden 24-GHz-Modellen benötigt der 77-GHz-Sensor nur ein Drittel der Antennenfläche, da die drei Trägerfrequenzen bei gleicher Auflösung das gleiche Sichtfeld abdecken können. Die dreifache Frequenz hilft dem System auch, die Geschwindigkeit zu messen, was zu einer dreifachen Genauigkeit der 24-GHz-Version führt.

Das MRR ist ein bistabiles Multimode-Radar mit vier unabhängigen Empfangskanälen und digitaler Beamforming (DBF). Dies macht es möglich, die Antenneneigenschaften des MRR so zu formen, so dass getrennte Antennen in verschiedenen Richtungen verwendet werden können. Daher kann das Sichtfeld des Radars (Field of View) abhängig von der Situation variabel verwendet werden, und die Genauigkeit der Winkelmessung kann verbessert werden. Durch Konzentration der Hauptantenne auf eine schmale Hauptkeule kann das Annäherungsverhalten von Fahrzeugen auf große Entfernungen (bis zu 160 Metern) bei höheren Fahrzeuggeschwindigkeiten detektiert werden, während die Beeinflussung von Fahrzeugen auf benachbarten Spuren verringert werden kann. Im Nahbereich erreicht die Verwendung

von Elevationsantennen einen weiten Öffnungswinkel. So können beispielsweise sich vorbeibewegende Fußgänger hinter geparkten Fahrzeugen frühzeitig entdeckt werden. Die unabhängige Messung der Elevationswinkel mittels einer Elevationsantenne ermöglicht eine starke Hindernisklassifikation, die auch bei stationären Objekten zu einer zuverlässigen Bremsreaktion führt. Abbildung 2-6 zeigt die Sichtfelder der zwei Antennen.



**Abbildung 2-6: Field of View der Antennen [12]**

Bosch hat das MRR in zwei Typen entwickelt, nämlich Front- und Heckvariante. Die Heckvariante des MRR überwacht den hinteren Bereich des Fahrzeugs. Auf diese Weise können die Fahrzeuge im toten Winkel und sich dem Fahrzeug von hinten nähernden Objekten genau identifiziert werden. Die technischen Merkmale des Sensors sind in Tabelle 2.1 angezeigt.

**Tabelle 2.1: Technische Merkmale des MRRs**

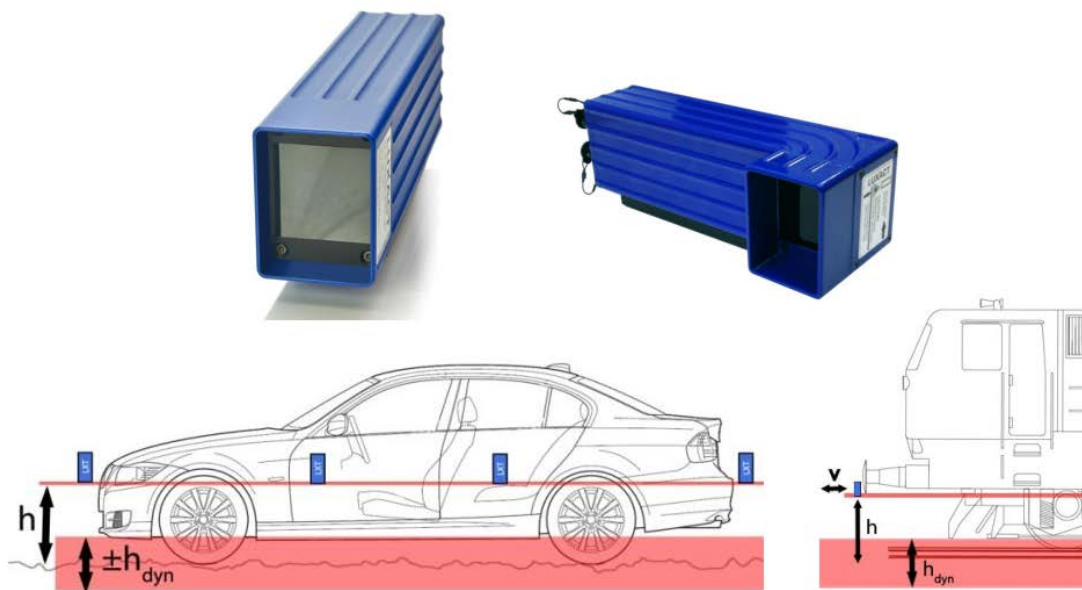
	MRR Front	MRR Heck
<b>Frequenzbereich</b>	76 ... 77 GHz	76 ... 77 GHz
<b>Erkennungsbereich</b>	0,36 ... 160 m	0,36 ... 80m
<b>Sichtfeld (horizontal)</b>		
<b>Hauptantenne</b>	± 6° (160 m)	± 5° (70 m)
	± 9° (100 m)	
	± 10° (60 m)	± 75° (Nahbereich)
<b>Elevationsantenne</b>	± 25° (36 m)	
	± 42° (12 m)	
<b>Messgenauigkeit</b>		
<b>Abstand</b>	0,12 m	0,12 m
<b>Geschwindigkeit</b>	0,11 m/s	0,14 m/s
<b>Winkel</b>	± 0,3°	± 0,8°
<b>Zykluszeit</b>	~ 60 ms	~ 60 ms
<b>Modulation</b>	Frequenzmodulation (FMCW)	

### 2.2.2 Optische Sensoren

Grundsätzlich arbeiten optische Sensoren nach den gleichen Prinzipien wie beim Radar, nur statt der Radiowellen werden Lichtstrahlen verwendet.

#### SMG Luxact Optischer Sensor

Luxact ist ein hochpräzises Messsystem von SMG Technologie GmbH zur berührungslosen Erfassung von Geschwindigkeiten, Wegen und ggf. Beschleunigungen von fahrenden Objekten, wie Kraftfahrzeugen, Zügen, Landmaschinen etc. bezüglich einer Oberfläche. Der Sensor misst seine relative Geschwindigkeit in Bezug auf eine ggf. irreguläre, unebene, abwechselnde Oberflächen und auch einige extra Parameter. Dank der einzigartigen Eigenschaften der Luxact-Technologie kann die Oberfläche frei gewählt werden, da die Eigenschaften der gegenüberliegenden Oberfläche und ihr Abstand vom Sensorkopf die Messqualität und Genauigkeit nicht wesentlich beeinflussen. In Abbildung 2-7 werden neben dem Aussehen der Sensoren auch die Anwendungsmöglichkeiten sowohl am Auto als auch auf einem Schienenfahrzeug dargestellt.



**Abbildung 2-7: Vertikale und Horizontale Version von Luxact (vgl. [13])**

LUXACT konzentriert sich auf die makroskopische Struktur der Oberfläche. Der Fluss des integrierten Senders wird von der streuenden Oberfläche in das optische System reflektiert, wo die Fluktuation der Intensität ausgewertet wird. Die Frequenz des Reflexions- und Intensi-

tätsmodulationsflusses ist dann proportional zur Relativgeschwindigkeit des Sensors auf der Streufläche. Dank des nicht blendenden optischen Systems kann das Sensorarray darüber hinaus unter allen Licht- und Oberflächenbedingungen ausreichend Licht erhalten und so ein bemerkenswertes Signal-Rausch-Verhältnis und eine hohe Messrobustheit auch unter schlechten Bedingungen gewährleisten.

Neben optischen Messmethoden hat die Technologie von Luxact auch eine weitere unverzichtbare und wichtige Eigenschaft, nämlich ein genaues Inertialsystem. Durch die Synergie der beiden Messmethoden kann auch die Verfügbarkeit und Zuverlässigkeit der Messung unter erschwerten Bedingungen deutlich verbessert werden. [13] Die technischen Daten bzw. die Beschreibungen der Messgenauigkeiten werden in Tabelle 2.2 angezeigt.

**Tabelle 2.2: Technische Daten von Luxact [14]**

LUXACT Versionen	Low-Speed	Standard	High-Speed
Messbereich Geschwindigkeit	0,08-30 km/h	0,2-300 km/h	0,5-500 km/h
Auflösung Geschwindigkeit	0,01 km/h	0,01 km/h	0,01 km/h
Auflösung Weg	0,01 m	0,01 m	0,01 m
Messunsicherheit Geschwindigkeit	$\leq 0,5 \%FS RMS$	$\leq 0,2 \%FS RMS$	$\leq 0,1 \%FS RMS$
	$\leq 0,25 \%typ.$	$\leq 0,1 \%typ.$	$\leq 0,05 \%typ.$
Messunsicherheit Weg bei $s > 200m$	$\leq 0,2 \%$	$\leq 0,1 \%$	$\leq 0,1 \%$
Messrate	50 Hz	50 / 250* / 1000* Hz	50 / 250* / 1000* Hz
Nominaler Arbeitsabstand	300* / 600 / 900* mm	300* / 600 / 900* mm	300* / 600 / 900* mm
Frequenzbandbreite von Geschwindigkeit (Messrate)	15 Hz @ 50 Hz	15 Hz @ 50 Hz	15 Hz @ 50 Hz
	75 Hz @ 250 Hz	75 Hz @ 250 Hz	75 Hz @ 250 Hz
	300 Hz @ 1000 Hz	300 Hz @ 1000 Hz	300 Hz @ 1000 Hz
Dynamischer Arbeitsabstand	$\pm 30 \%$ von h	$\pm 30 \%$ von h	$\pm 30 \%$ von h
Lichtquelle	Halogen oder LED*	Halogen oder LED*	Halogen oder LED*

### 2.2.3 Drehzahlbasierte Sensoren

Drehzahlbasierte Sensoren gehören zu den Weg-Zeit-Messung basierten Geschwindigkeits-sensoren, die ebenfalls die Zeit messen, die ein Objekt benötigt, um eine festgelegte Strecke zurückzulegen (vgl. [5]). Diese Methode ist auf die Verwendung auf Rädern oder ähnlichen rollenden Objekten beschränkt. Der sogenannte festgelegte Abstand bezieht sich hiermit auf die Länge des Rollens. Das heißt, die zurückgelegte Strecke eines Objekts pro Rotation seiner Räder entspricht dem Abrollumfang des Rads  $U$ :

$$U = 2\pi \cdot R \quad (2.7)$$

In der Formel ist  $R$  der dynamische Rollradius des Rades. Obwohl die Form des Reifengürtels flexibel ist, behält es seine Länge bei, wenn es entfaltet wird, ändert sich der Rollkreis jedoch aufgrund der Belastung und der Geschwindigkeit. Der Verschleiß des Reifens oder sogar ein Anstieg oder Abfall des Luftdrucks können auch den Abrollumfang des Rades verändern. Bei Schienenfahrzeugen kann der durch den Verschleiß des Rades verursachte Messfehler 10% erreichen. Wenn größere oder kleinere Räder montiert werden, ändert sich in jedem Fall der Wert des Bildlaufkreises. Die Anzahl der Radumdrehungen pro Zeiteinheit wird als Drehzahl  $n$  oder als Frequenz bezeichnet. Die mittlere Geschwindigkeit kann dann aus der Drehzahl und dem Abrollumfang abgeleitet werden:

$$v = n \cdot 2\pi \cdot R \quad (2.8)$$

Der Schlupf des Rades führt zu einem falschen Messergebnis. Wenn beispielsweise der Zug stark bremst, wird die von den Rädern im verriegelten Zustand zurückgelegte Entfernung nicht berücksichtigt. Die von der Geschwindigkeit gemessene Geschwindigkeit ist daher niedriger als der tatsächliche Wert. Bei starker Beschleunigung führt der schnelle Schlupf des Rades auch zu einer Abweichung vom Messergebnis. In praktischen Anwendungen wird die Geschwindigkeit durch den entsprechenden Sensor direkt am Rad, an der Achse oder am Getriebe gemessen. Und je nach Anwendung gibt es unterschiedliche Messmethoden. Zum Beispiel werden die auf der magnetischen Induktion basierende Inkrementalgeber zunehmend verwendet.

Das in Straßen- und Schienenfahrzeugen eingesetzte Tachometer ist eine der bekanntesten Anwendungen, die auf die Messung von Drehzahl basiert. Das Tachometer zeigt dem Fahrer die aktuelle Fahrzeuggeschwindigkeit an und berechnet gleichzeitig den Kilometerstand.

**Baumer BPIK Achsimpulsgeber [15]**

Als zuverlässiger Partner im Schienenverkehr hat die Baumer Group seit vielen Jahren verschiedene Sensortechnologien für die hohen Anforderungen der Branche bereitgestellt. Der neue hoch-performante Achsimpulsgeber BPIK (siehe Abbildung 2-8) ist das Ergebnis der Kombination aus langjähriger Berufserfahrung in Schienenfahrzeugen und den neuesten technologischen Fortschritten.



**Abbildung 2-8: Lagerloser Achsimpulsgeber Baumer BPIK [15]**

Mit Hilfe eines magnetischen Polrads kann der lagerlose Achsimpulsgeber Drehzahlen mit hoher Auflösung messen und für eine Vielzahl von Anwendungen an Fahrzeugdrehgestellen eingesetzt werden. Um sich den härtesten Bedingungen an der Achse anpassen zu können, stehen bis zu 1200 Impulse pro Umdrehung zur Verfügung. Der Impulsgeber ist in einem robusten Aufbau- und Messprinzip mit extrem geringem Jitter der Signale entwickelt. Dank seines rauscharmen Signals kann der Sensor die Geschwindigkeit des Rades sehr genau messen.

Wie die anderen lagerlosen Drehgeber von Baumer sind die Betriebsbedingungen von BPIK nahezu verschleißfrei und das Produkt ist besonders unempfindlich gegenüber Vibrationen und Verunreinigungen. Auf Basis aller wichtigen Bahnnormen bietet das umfangreiche Produktportfolio des Konzerns eine geeignete und zuverlässige Lösung für die Drehzahlmessung auf Drehgestellen.

### 2.2.4 Satellitengestützte Sensoren

Für die satellitengestützten Sensoren gibt es drei mögliche Verfahren [5]:

1. Die zurückgelegte Strecke während eines bestimmten Zeitintervalls wird durch eine periodische Positionsmessung berechnet, woraus die Durchschnittsgeschwindigkeit berechnet wird. Diese Methode kann eine Genauigkeit von einigen Metern pro Sekunde erreichen.
2. Gemäß dem Doppler-Effekt führt die relative Bewegung zwischen dem Satelliten und dem (bewegten) Empfänger zu einer Verschiebung der Trägerfrequenz der Satellitensignale. Durch das Vergleichen der Frequenz des empfangenen Satellitensignals mit dem am Empfänger erzeugten Signal kann die Dopplerverschiebung berechnet werden und daraus die Bewegungsgeschwindigkeit des Objekts abgeleitet werden. Auf diese Weise kann eine Genauigkeit von einigen Zentimetern pro Sekunde erreicht werden.
3. Time-differenced-Carrier-Phase(TDCP)-basierte Geschwindigkeitsmessung. Dieses Verfahren misst die Phasenverschiebung, die durch die relative Bewegung zwischen dem empfangenden Satellitensignal und dem (bewegten) Empfänger erzeugt wird. Die Dopplerverschiebung der Trägerfrequenz des Satellitensignals wird durch das Messen der Phasendifferenz zwischen zwei aufeinanderfolgenden Zyklen bestimmt. Daraus kann wiederum auf die Geschwindigkeit des Objekts geschlossen werden. Im Vergleich zu der zweiten Methode ist das Rauschen bei der Messung der Phasenverschiebung deutlich geringer als das Rauschen bei der direkten Messung der Frequenzverschiebung. Dieses Verfahren kann Geschwindigkeiten mit einer Genauigkeit von einigen Millimetern pro Sekunde messen.

Während das erste Verfahren zu den Weg-Zeit-Messung basierten Verfahren gehört, gehören die beiden letztgenannten Verfahren zu den Doppler-Effekt basierten Verfahren.

Das weltweit bekannteste und am längsten laufende globale Navigationssatellitensystem (GNSS) ist das vom US-Verteidigungsministerium in den 1970er Jahren entwickelte Global Positioning System (GPS). Um die Positionsinformation auf der angenommenen zweidimensionalen Erdoberfläche zu bestimmen und den Empfängeruhrenfehler zu korrigieren, ist es



notwendig, drei Satellitensignale zu empfangen (2D-Position-Fix). Um jedoch die Position im dreidimensionalen Raum zu bestimmen (3D-Position-Fix), wird das Signal des vierten Satelliten benötigt. Mit dessen Hilfe ist es auch möglich, die Höhe des Empfängers über der Erdoberfläche zu bestimmen.

### **Dilution of Precision (DOP)**

Die Genauigkeit der Positionsbestimmung hängt von der Position der verbundenen Satelliten zueinander und der Position relativ zu dem zu bestimmenden Punkt ab. Natürlich beeinflusst dies auch die Genauigkeit der Zeitmessung bzw. die Genauigkeit der Laufzeit. Der effektivste Winkel zwischen Satellit A, Empfänger und Satellit B ist  $90^\circ$ . Wenn dieser Winkel sehr klein ist oder nahe bei  $180^\circ$  liegt, schneiden sich die Entfernungskreise bzw. -Kugeln in einem flacheren Winkel und die Messgenauigkeit ist geringer. Die Größe des DOP wird durch den Kehrwert der Fläche bzw. des Volumens bestimmt, das die Einheitsvektoren in Richtung der Satelliten aufspannen. [16]

Der DOP-Wert kann verwendet werden, um zu beschreiben, wie gut die Positionen der Satelliten sind. Der ideale Wert von eins steht für das höchstmögliche Konfidenzniveau. Mit einem DOP-Wert von eins bis zwei sollten alle anspruchsvollsten Messungen möglich sein. Um die Qualität der durchgeführten Messungen für die meisten Anwendungen noch ausreichend zu gewährleisten sollte der DOP-Wert kleiner als zehn sein. [17]

Jedes Gerät, das mit einem GNSS-Empfänger und entsprechender Software ausgestattet ist, kann die Geschwindigkeit basierend auf den obigen Verfahren messen. Obwohl es bei einigen Navigationsgeräten hochwertige Empfänger gibt, um genauere Geschwindigkeitsmessungen zu erreichen, wird auch oft von kommerziellen Herstellern von Navigationsgeräten nicht darauf hingewiesen, ob die gemessene Geschwindigkeit auf Basis der Weg-Zeit-Messung oder des Doppler-Effekts ermittelt wird. [5]

### Javad SIGMA-G3T

Auf dem Versuchsfahrzeug RailDrIVE® des Instituts für Verkehrssystemtechnik des Deutschen Zentrums für Luft- und Raumfahrt e.V. (DLR) Braunschweig wird das SIGMA-G3T (siehe Abbildung 2-9) von Javad GNSS Inc. eingesetzt.



**Abbildung 2-9: Javad SIGMA-G3T [18]**

SIGMA-G3T ist ein leistungsstarker Empfänger für hochgenaue Anwendungen wie Referenzstationen und Continuously Operating Reference Stations (CORS). Das Gerät kann auch als portable Basisstation für Echtzeit-Kinematik-Anwendungen (RTK) und als wissenschaftliche Station zur Sammlung von Informationen für spezielle Studien eingesetzt werden. Die Leistungsbeschreibungen sind in Tabelle 2.3 dargestellt.

**Tabelle 2.3: Leistungsbeschreibungen von Javad SIGMA-G3T [18]**

Performance Specifications	
<b>Autonomous</b>	< 2 m
<b>Static, Fast Static accuracy</b>	Horizontal: $0.3 \text{ cm} + 0.1 \text{ ppm} * \text{base\_line\_length}$
	Vertical: $0.35 \text{ cm} + 0.4 \text{ ppm} * \text{base\_line\_length}$
<b>Kinematic accuracy</b>	Horizontal: $1 \text{ cm} + 1 \text{ ppm} * \text{base\_line\_length}$
	Vertical: $1.5 \text{ cm} + 1.5 \text{ ppm} * \text{base\_line\_length}$
<b>RTK (OTF) accuracy</b>	Horizontal: $1 \text{ cm} + 1 \text{ ppm} * \text{base\_line\_length}$
	Vertical: $1.5 \text{ cm} + 1.5 \text{ ppm} * \text{base\_line\_length}$
<b>DGPS accuracy</b>	< 0.25 m post processing, < 0.5 m real-time
<b>Real-time heading accuracy</b>	$\sim 0.004/L \text{ [rad] RMS}$ , where L is the antenna separation in [m]
<b>Cold Start</b>	< 35 seconds
<b>Warm Start</b>	< 5 seconds
<b>Reacquisition</b>	< 1 second

## 2.3 CAN-Bus

Der CAN-Bus (Controller Area Network) ist ein zu den Feldbussen gehörendes serielles Bussystem und wurde von Bosch in der zweiten Hälfte der 80er Jahre entwickelt und seit 1991 als erstes Class C-Netz in Fahrzeugen eingesetzt [19]

Der treibende Faktor in der Automobilindustrie für die Entwicklung von CAN und anderen sogenannten Autobusprotokollen war die Notwendigkeit, neue Lösungen für das Problemfeld zu suchen, das sich mit dem steigenden Bedarf für Kommunikation in Kraftfahrzeugen ergab – unter der Randbedingung, die damit verbundene, drohende Erhöhung der Komplexität des Kabelbaums im Auto zu reduzieren. Die Verkabelung in Autos kann mehrere Kilometer Kabellänge umfassen und mehr als 100 kg schwer sein. Es ist leicht nachvollziehbar, dass solche Kabelbäume schwierig zu installieren sind. [20] Obwohl die erste CAN-Implementierung von Bosch in Kooperation mit Intel durchgeführt wurde, hat Bosch jedoch frühzeitig eine offene Lizenzpolitik betrieben, die es jedem Mikrocontroller-Hersteller ermöglichte, CAN-Module anzubieten. Damit die CAN-Controller der verschiedenen Hersteller miteinander kompatibel kommunizieren zu können, hat Bosch auch ein Referenzmodell und ein Testprogramm zur Verfügung gestellt.

Der ISO 11898-2 zum CAN-ISO-Standard definiert für Applikationen mit Bitraten  $\geq 250\text{ kbit/s}$  (High Speed CAN) die Verwendung einer nach Möglichkeit verdrehten Zwei-Draht-Leitung als echter Linien-Bus mit Stichleitungen von max. 30 cm zu den einzelnen Steuergeräten. Der Bus muss an beiden Enden mit dem Wellenwiderstand der Zwei-Draht-Leitung, typischerweise ca.  $120\ \Omega$ , abgeschlossen werden (siehe Abbildung 2-10). [21]

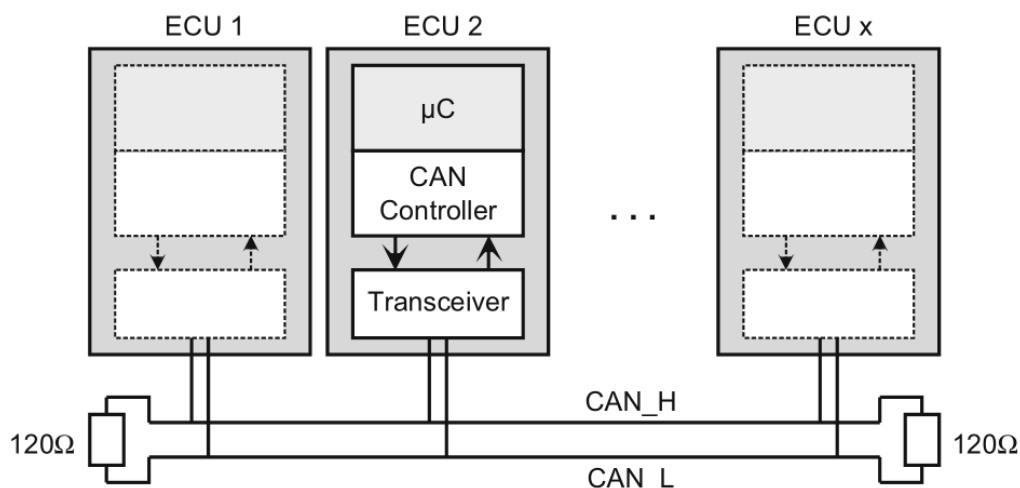


Abbildung 2-10: High-Speed-CAN-Bus [21]

### 2.3.1 SocketCAN unter Linux

Das SocketCAN-Paket ist eine Implementierung von CAN-Protokollen (Controller Area Network) für Linux, die von der Konzernforschung der Volkswagen AG zum Linux-Kernel als Open Source beigestellt wurde. SocketCAN ist eine offizielle CAN-API des Linux-Kernels [22] und auch bekannt als Low Level CAN Framework (LLCF). Während es auf Basis von Zeichengeräten andere CAN-Implementierungen für Linux gab, verwendet SocketCAN die Berkeley-Socket-API, den Linux-Netzwerkstack und implementiert die CAN-Gerätetreiber als Netzwerkschnittstellen. Die CAN-Socket-API wurde so ähnlich wie möglich zu den TCP / IP-Protokollen entwickelt, um Programmierern, die mit der Netzwerkprogrammierung vertraut sind, die Verwendung von CAN-Sockets zu erleichtern.

Die meisten vorherrschenden Implementierungen werden als Gerätetreiber für einige CAN-Hardware verwendet, die auf Zeichengeräten basieren und vergleichsweise wenig Funktionalität bieten. Normalerweise gibt es nur einen hardwarespezifischen Gerätetreiber, der eine Zeichengeräteschnittstelle zum Senden und Empfangen von RAW-CAN-Frames direkt zu oder von der Steuerungshardware bereitstellt. Außerdem unterstützen die meisten Implementierungen von Zeichengeräten nur einen einzigen Prozess, um das Gerät zur gleichen Zeit zu öffnen, ähnlich wie bei einer seriellen Schnittstelle. Der Austausch des CAN-Controllers erfordert die Verwendung eines anderen Gerätetreibers und häufig die Anpassung großer Teile der Anwendung an die neue Treiber-API. [23] SocketCAN wurde entwickelt, um alle diese Einschränkungen zu überwinden. Zudem wurde eine neue Protokollfamilie genannt PF\_CAN hinzugefügt, welches neben anderen bekannten Protokoll-Familien wie z. B. PF\_INET für das Internet unabhängig koexistiert. Ein Gerätetreiber für die CAN-Controller-Hardware registriert sich bei der Linux-Netzwerkschicht als Netzwerkgerät, so dass CAN-Frames vom Controller an die Netzwerkschicht und weiter an das CAN-Protokollfamilienmodul übergeben werden können. Außerdem stellt das Protokollfamilienmodul eine API für die zu registrierenden Transportprotokollmodule bereit, so dass eine beliebige Anzahl von Transportprotokollen dynamisch geladen oder entladen werden kann.

Die Anwendung von SocketCAN richtet zunächst den Zugang der CAN-Schnittstelle durch Initialisierung eines Sockets (ähnlich einer TCP/IP-Kommunikation) ein.

```
s = socket(PF_CAN, SOCK_RAW, CAN_RAW);
```

Nach der erfolgreichen Erstellung des Sockets wird normalerweise der Systemaufruf `bind()` verwendet, um den Socket an eine CAN-Schnittstelle zu binden. Bei erfolgreicher Anbindung, ist der Socket durch z.B. die Funktionen fürs Lesen `read()` und Schreiben `write()` nutzbar.

Die grundlegende CAN-Frame- und CAN-FD-Frame-Struktur sind in `include/linux/can.h` definiert:

```
struct can_frame {
    canid_t can_id; /* 32 bit CAN_ID + EFF/RTR/ERR flags */
    __u8 can_dlc; /* frame payload length in byte (0 ..
CAN_MAX_DLEN) */
    __u8 data[CAN_MAX_DLEN] __attribute__((aligned(8)));
};

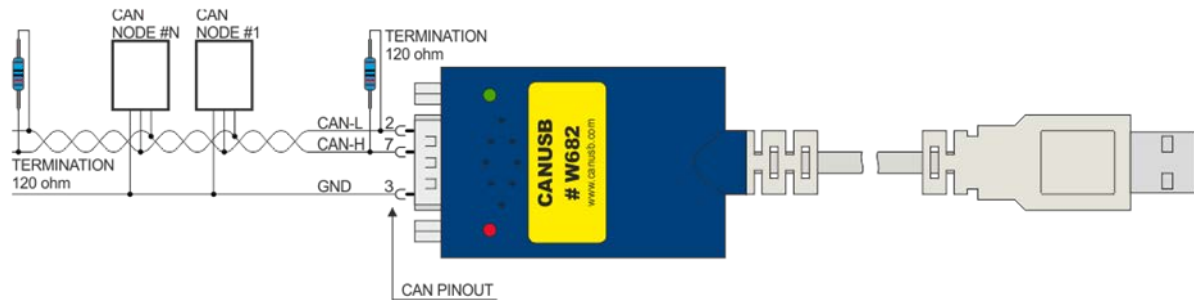
struct canfd_frame {
    canid_t can_id; /* 32 bit CAN_ID + EFF/RTR/ERR flags */
    __u8 len; /* frame payload length in byte */
    __u8 flags; /* additional flags for CAN FD */
    __u8 __res0; /* reserved / padding */
    __u8 __res1; /* reserved / padding */
    __u8 data[CANFD_MAX_DLEN] __attribute__((aligned(8)));
};
```

Die `can_id`, die Payload-Länge und die Payload-Daten von `canfd_frame` und `can_frame` besetzen das gleiche Offset innerhalb der jeweiligen Strukturen. Dies ermöglicht es, die verschiedenen Strukturen sehr ähnlich zu behandeln. Wenn der Inhalt einer `can_frame` Struktur in eine `canfd_frame` Struktur kopiert wird, können alle Strukturelemente so wie sie sind verwendet werden, nur die `data[]` Strukturelemente werden erweitert.

Die Ausrichtung des Payloads auf eine 64-Bit-Grenze ermöglicht es dem Benutzer, seine eigenen Strukturen zu definieren, um einfach auf die CAN-Payload zuzugreifen. Am CAN-Bus ist standardmäßig keine Byte-Reihenfolge angegeben. Durch einen `read()` Systemaufruf an einem `CAN_RAW`-Socket kann eine `can_frame` Struktur an den Benutzerbereich übertragen werden.

### 2.3.2 CANUSB

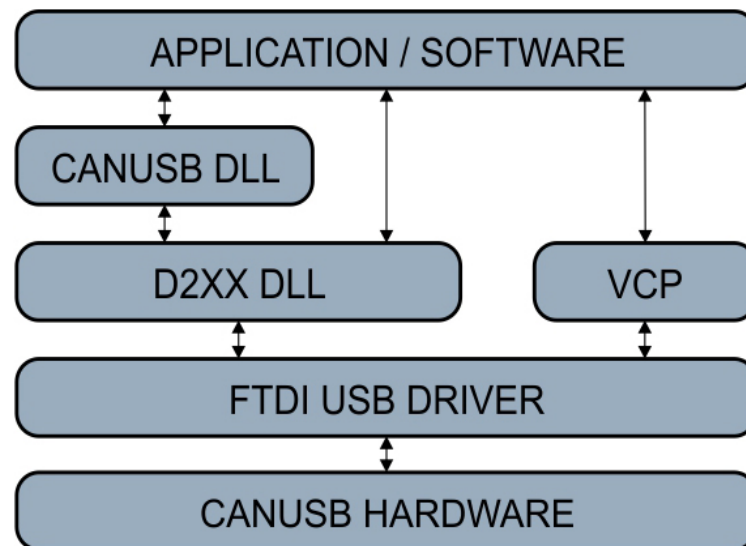
CANUSB aus Lawicel ist ein intelligenter Schnittstellenwandler zur Umsetzung von CAN-Signalen. Der kleine Adapter kann einfach an jeden USB-Port eines PC mit verschiedenen Betriebssystemen wie z.B. Windows, Linux etc., angeschlossen werden. [24] Das Verbindungsbeispiel seitens der CAN-Schnittstelle wird in der Abbildung 2-11 gezeigt.



**Abbildung 2-11: CANUSB mit dem CAN-Bus verbunden [24]**

Nachdem CANUSB seit vielen Jahren in zahlreichen Kundenprojekten angewendet wurde, hat sich CANUSB zu einer sogenannten De-facto-Standard Schnittstelle für CAN-Kommunikation am PC entwickelt. Dank der nachhaltigen Produktwartung vom schwedischen Herstellers Lawicel, bietet CANUSB heutzutage eine ausgereifte, einfach zu bedienende CAN-to-USB-Lösung.

Es stehen zwei Treibermodelle zur Verfügung. Im VCP- (Virtual COM Port) Modus wird keine spezielle Treibersoftware mehr benötigt, wenn ein serieller Port und ein einfaches ASCII-Protokoll verwendet werden. Im D2XX-Modus dagegen, muss eine spezielle DLL (Dynamic Link Library) vorbereitet werden. Die CAN-Messages können damit im Standard-ASCII-Format gesendet bzw. empfangen werden. Die Abbildung 2-12 stellt ein Diagramm der Treibermodelle von CANUSB dar. Man kann sehen, dass es einfacher ist, das Gerät im VCP-Modus als im D2XX-Modus anzuwenden.



**Abbildung 2-12: Diagramm der Treibermodelle von CANUSB [25]**

Um CANUSB zu benutzen muss der FTDI-USB-Treiber zuerst installiert werden, der unter Linux ab Kernel 3.0.0-19 schon enthalten ist. Und um dennoch den eingebauten COM-Port zu verifizieren, wird durch Eingabe von

```
dmesg | grep FTDI
```

im Linux-Terminal folgende Ausgabe angezeigt. [26]

```
[10170.987708] USB Serial support registered for FTDI USB Serial Device
[10170.987915] ftdi_sio 9-1:1.0: FTDI USB Serial Device converter detected
[10170.991172] usb 9-1: FTDI USB Serial Device converter now attached to ttyUSB0
[10170.991219] ftdi_sio: v1.6.0:USB FTDI Serial Converters Driver
```

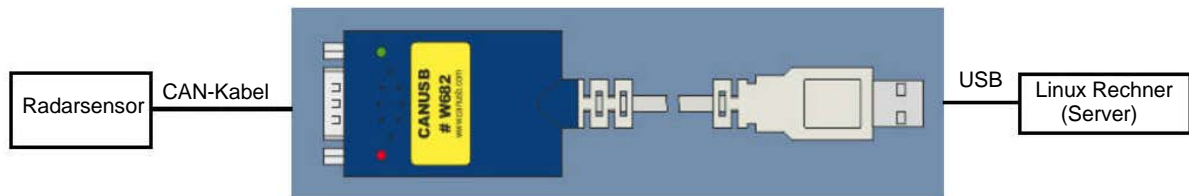
Die gezeigte Abkürzung „ftdi\_sio“ steht für „Future Technology Devices International\_ Serial Input/Output“. Das bedeutet, dass der Treiber schon initial im Betriebssystem vorhanden ist. Außerdem wird durch „usb 9-1: FTDI USB Serial Device converter now attached to ttyUSB0“ noch darauf hingewiesen, dass das CANUSB von dem Betriebssystem mit dem Gerätenamen „ttyUSB0“ erkannt wird.



### 3 Vorbereitung und Durchführung von Messfahrten

#### 3.1 Verbindung des CANUSB

Der Rechner, der CANUSB und der Radarsensor bzw. andere CAN-Geräte werden, wie in der Abbildung 3-1 dargestellt, miteinander verbunden. Auf einer Seite verbindet sich der CANUSB mit dem CAN-Gerät durch die CAN-Schnittstelle. Auf der anderen Seite kann der CANUSB an einen beliebigen USB-Port des Rechners angeschlossen werden.



**Abbildung 3-1: Verbindungsdiagramm**

Der am Rechner angeschlossenen CANUSB wird durch das Betriebssystem (z.B. Ubuntu) unter dem Pfad „/dev“ als ein ttyUSB-Gerät erkannt.

SLCAN steht für "Serial CAN Protocol" und ist ein Typ der durch SocketCAN angebotenen CAN-Schnittstellen, welches ein einfaches ASCII-Protokoll für die Kommunikation mit USB-CAN-Adaptern ist. Speziell für kostengünstige CAN-Adapter ist dieses Protokoll zum Standard geworden. SLCAN ermöglicht es, das erkannte ttyUSB-Gerät als ein Netzwerkgerät behandeln zu können. Voraussetzung ist, dass die drei relevanten Kernel-Module „can“, „can\_raw“ und „slcan“ bereits geladen sind. Mit Hilfe von dem Kommando in Abbildung 3-2 kann der Status überprüft werden.

```

raildrive@raildrive-Latitude-E6530:~$ lsmod | grep can
Module                Size  Used by
slcan                  16384  0
can_raw                20480  0
can                    45056  1 can_raw
  
```

**Abbildung 3-2: Überprüfung des Status der Kernel-Module**

Wenn die drei Module noch nicht geladen sind, müssen sie manuell gestartet werden. [27]

```
sudo modprobe can
sudo modprobe can_raw
sudo modprobe slcan
```

Zur Behandlung der Schnittstelle wird zuerst ein spezifischer Daemon gebraucht, der in den Werkzeugen für SocketCAN (can-utils) enthalten ist.

```
sudo apt-get install can-utils
```

Anschließend kann diese serielle Schnittstelle mit einem virtuellen CAN-Gerät verbunden werden.

```
sudo slcand -o -c -f -sX /dev/ttyUSBX slcan0
```

Hierfür muss „-sX“ je nach der hardwarespezifischen CAN-Bitrate angepasst werden (siehe Tabelle 3.1).

**Tabelle 3.1: Einstellung der CAN-Bitrate [28]**

sX	standard CAN-Bitrate
s0	10 Kbit/s
s1	20 Kbit/s
s2	50 Kbit/s
s3	100 Kbit/s
s4	125 Kbit/s
s5	250 Kbit/s
s6	500 Kbit/s
s7	800 Kbit/s
s8	1000 Kbit/s

Die Schnittstelle wird dann zum System hinzugefügt. Dabei ist der Anfangszustand „DOWN“ (siehe Abbildung 3-3).

```

raildrive@raildrive-Latitude-E6530:~$ ip addr
1: lo: <LOOPBACK,UP,LOWER UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: eth0: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc pfifo_fast state DOWN group default qlen 1000
    link/ether d4:be:d9:70:64:be brd ff:ff:ff:ff:ff:ff
3: wlan0: <BROADCAST,MULTICAST,UP,LOWER UP> mtu 1500 qdisc mq state UP group default qlen 1000
    link/ether 24:77:03:be:2b:60 brd ff:ff:ff:ff:ff:ff
    inet 10.17.247.34/20 brd 10.17.255.255 scope global wlan0
        valid_lft forever preferred_lft forever
    inet6 fe80::2677:3ff:febe:2b60/64 scope link
        valid_lft forever preferred_lft forever
4: slcan0: <NOARP> mtu 16 qdisc noop state DOWN group default qlen 10
    link/can

```

**Abbildung 3-3: Liste der Netzwerkschnittstellen**

Um die CAN-Messages einzulesen muss durch folgendes Kommando die Schnittstelle aktiviert werden.

```
sudo ifconfig slcan0 up
```

Der Status ändert sich dann von „DOWN“ auf „UNKNOWN“. Durch die richtige Stromversorgung des Sensors kann mit Hilfe des Werkzeuges „candump“ überprüft werden, ob die Kommunikation erfolgreich stattfindet (siehe Abbildung 3-4). Darüber hinaus leuchtet die grüne LED am CANUSB, immer wenn eine CAN-Nachricht empfangen oder gesendet wird.

```

raildrive@raildrive-Latitude-E6530:~$ candump slcan0
slcan0 3F2 [8] 00 00 B7 07 D4 30 00 80
slcan0 3F3 [8] 00 60 02 FF 3A 43 10 F0
slcan0 3F7 [8] 00 20 07 FF D2 03 8B FF
slcan0 3F5 [8] 00 00 00 00 00 00 00 80
slcan0 3F9 [8] 00 00 00 00 00 00 00 80
slcan0 3FB [8] 00 00 00 00 00 00 00 80
slcan0 3FF [8] 00 00 00 00 00 00 00 80
slcan0 3FD [8] 00 00 00 00 00 00 00 80
slcan0 401 [8] 00 00 00 00 00 00 00 80
slcan0 403 [8] 00 00 00 00 00 00 00 80
slcan0 407 [8] 01 90 09 FF 80 46 C8 FF
slcan0 405 [8] 00 00 00 00 00 00 00 80

```

**Abbildung 3-4: Anzeige der CAN-Messages im Terminal**

## 3.2 Programme zur Speicherung und Dekodierung von CAN-Messages

In der Abbildung 3-4 zeigt die erste Spalte „slcan0“ den Name der eingelesenen Schnittstelle. In der zweiten Spalte werden die CAN-IDs aufgelistet, die dazu dienen, die Spezifikation jeweiliger CAN-Messages zu identifizieren. Die Ziffer „8“ zwischen den eckigen Klammern in der dritten Spalte weist darauf hin, dass die Länge der nachfolgenden Dateninhalte acht Bytes beträgt. Um die Dateninhalte richtig dekodieren zu können, muss das Format der Byte-Reihenfolge bekannt sein.

### Byte-Reihenfolge

Die Byte-Reihenfolge (englisch endianness) bezieht sich auf die sequenzielle Reihenfolge, in der Bytes zu größeren numerischen Werten angeordnet werden, wenn sie im Speicher gespeichert sind oder wenn sie über digitale Verbindungen übertragen werden. Endianness ist in der Informatik von Interesse, weil zwei widersprüchliche und inkompatible Formate gebräuchlich sind: Wörter können im Big-Endian- oder Little-Endian-Format dargestellt werden, abhängig davon, ob Bytes vom Groß-Ende (höchstwertiges Byte) oder vom Klein-Ende (niedrigstwertiges Byte) aus geordnet sind. [29]

Im Big-Endian-Format wird das höchstwertige Byte, das das höchstwertige Bit enthält, zuerst gespeichert (besitzt die niedrigste Adresse) oder zuerst gesendet. Dann werden die folgenden Bytes in abnehmender Signifikanzreihenfolge gespeichert oder gesendet, wobei das niedrigstwertige Byte, das das niedrigstwertige Bit enthält, zuletzt gespeichert (mit der höchsten Adresse) oder zuletzt gesendet wurde.

Das Little-Endian-Format kehrt diese Reihenfolge um: Das niedrigstwertige Byte wird zuerst gesendet oder gespeichert (mit niedrigsten Adresse) und das höchstwertige Byte zuletzt (mit höchsten Adresse). (siehe Tabelle 3.2)

Die beiden unterschiedlichen Arten der Datenspeicherung werden aus historischen Gründen allgemein als Intel-Format (Little-Endian) und Motorola-Format (Big-Endian) bezeichnet. Die meisten Computersysteme benutzen ein einziges Format für alle ihre Daten; das systemeigene Format wird automatisch verwendet. Beim Lesen von Speicher oder beim Empfangen

von übertragenen Daten von einem anderen Computersystem ist es jedoch oft erforderlich, Daten zwischen dem bevorzugten nativen Endian-Format und dem entgegengesetzten Format zu verarbeiten und zu übersetzen.

**Tabelle 3.2: Byte-Reihenfolge in Motorola- und Intel-Format [30]**

Bit Nummerierung von CAN-Daten in Motorola-Format								
Byte Nr.	0	1	2	3	4	5	6	7
Bit Nr.	63...56	55...48	47...40	39...32	31...24	23...16	15...8	7...0
Bit Nummerierung von CAN-Daten in Intel-Format								
Byte Nr.	0	1	2	3	4	5	6	7
Bit Nr.	7...0	15...8	23...16	31...24	39...32	47...40	55...48	63...56

Wegen der Geheimhaltungsvereinbarung zwischen DLR e.V. Braunschweig und dem Zulieferer des Automotive-Radarsensors (nachstehend „MRR-Sensor“ genannt) wird hierfür die Struktur der CAN-Messages nicht von dem MRR-Sensor, sondern von einem vorhandenen optischen Sensor (Luxact) analysiert und anschließend das zu implementierende Dekodierungsprogramm erklärt.

Gleich wie der eingesetzte MRR-Sensor wird die Kommunikation des Luxact mit dem Rechner ebenfalls durch einen CAN-Bus realisiert. Aber im Gegensatz zu den CAN-Messages vom MRR-Sensor sind nur die CAN-ID und die Struktur der nachfolgenden 8-Bytes Dateninhalte unterschiedlich, deswegen ist der zu Grunde liegende Algorithmus um die CAN-Daten zu dekodieren identisch. Darüber hinaus verwenden sowohl Luxact-Sensor als auch MRR-Sensor die Byte-Reihenfolge der CAN-Daten vom Intel-Format.

Obwohl beim Luxact-Sensor drei Typen von CAN-Messages, jeweils mit ID13, ID15 und ID17, vorhanden sind, sind alle Hauptsignale wie z.B. Geschwindigkeit und Abstand nur in der Message mit ID13 enthalten. In Tabelle 3.3 werden die Bitsbelegungen der Signale bzw. deren Spezifikationen dargestellt.

Tabelle 3.3: Signale in der Message mit ID13

Signals within message ID13: Main signals								
Signal Name	Type	Format	Mode	Start Bit	Length	Factor	Offset	Unit
AbsVelocity	Unsigned	Intel	Normal	0	15	0.01	0	km/h
SignVelocity	Unsigned	Intel	Normal	15	1	1	0	
Distance	Signed	Intel	Normal	16	32	0.001	0	m
Acceleration	Signed	Intel	Normal	48	16	0.01	0	m/(s <sup>2</sup> )
Signal name	Description							
AbsVelocity	unsigned speed in km/h, resolution 0.01 km/h							
SignVelocity	speed sign bit							
Distance	Distance signed in m, can be positive and negative, resolution 0.001m (driving 1 km back and 1 km forward would result in 0 km)							
Acceleration	Acceleration in m/s <sup>2</sup> , resolution 0.01 m/s <sup>2</sup> . Raw accelerations from accelerometer without any compensation							

Da die Daten im Intel-Format (Little-Endian) gesendet werden, muss zuerst die Reihenfolge der byteweise empfangenen CAN-Daten im Programm neu angepasst werden, um damit die Signale anhand der entsprechend Bitsbelegungen richtig zu dekodieren. Das heißt, dass das zuerst empfangene Datenbyte auf das niedrigstwertige Byte gestellt werden soll und das zuletzt empfangene Datenbyte auf das höchstwertige Byte.

```
// store 8 single Bytes as one 64bits unsigned int
uint64_t can_msg = 0x0000000000000000;

// allocate a 64-bits buff
uint64_t msg_buf64 = 0x0000000000000000;

for (i = 0; i < len; i++) // len is CAN Data length, 8 Bytes
{
    // change data[i] sequences of the can_frame
    can_msg = can_msg | ((msg_buf64 | cf->data[i]) << (i*8));
}
```

Anschließend können die Signale jeweils mit Hilfe einer geeigneten „Maske“ dekodiert bzw. deren Werte errechnet werden. Die `fprintf()` Funktion kann dazu dienen, die CAN-Daten und die ausgerechneten Werte der Signale zusammen mit dem jeweiligen entsprechenden Zeitstempel formatiert in zwei separaten Dateien zu speichern.

```
// unsigned, from bit 0 to bit 14, length 15, factor 0.01, unit km/h
uint16_t AbsVelocity;
// unsigned, bit 15, length 1, factor 1, speed sign bit
uint8_t SignVelocity;
// signed, bit 16 to bit 47, length 32, factor 0.001, unit m
int32_t Distance;
// signed, bit 48 to bit 63, length 16, factor 0.01, unit m/s2
int16_t Acceleration;

if (cf->can_id == 13) {
AbsVelocity = can_msg & 0x7fff;
SignVelocity = ( can_msg & 0x8000 ) >> 15;
Distance = ( can_msg & 0xffffffff0000 ) >> 16;
Acceleration = ( can_msg & 0xffff000000000000 ) >> 48;

// store the value of all signals in file
fprintf(logfile_text, "%010ld%03ld\t%d\t%.2f\t%d\t%.3f\t%.2f\n",
tv.tv_sec, tv.tv_usec/1000,
cf->can_id,
AbsVelocity * 0.01, // unit km/h
SignVelocity * 1,
Distance * 0.001, // unit m
Acceleration * 0.01 ); // unit m/s²

// store the 64bits Message in file
fprintf(logfile_data, "%010ld%03ld\t%02x%02x%02x%02x%02x%02x%02x%02x\n",
tv.tv_sec, tv.tv_usec/1000,
cf->can_id, cf->data[7], cf->data[6], cf->data[5], cf->data[4],
cf->data[3], cf->data[2], cf->data[1], cf->data[0]);
}
}
```

Wenn beispielsweise eine Reihe von CAN-Message aus Luxact-Sensor „0d0023ffffdb690319“ eingelesen, werden die Inhalte dieser Message wie in Tabelle 3.4 dekodiert bzw. gerechnet.

**Tabelle 3.4: Inhalte der CAN-Message mit ID13**

Signal Name	Type	Start Bit	Length	Hex	Factor	Value	Unit
<b>AbsVelocity</b>	Unsigned	0	15	319	0.01	Jul 93	km/h
<b>SignVelocity</b>	Unsigned	15	1	0	1	0	
<b>Distance</b>	Signed	16	32	ffffdb69	0.001	-9.367	m
<b>Acceleration</b>	Signed	48	16	23	0.01	0.35	m/(s <sup>2</sup> )

### **3.3 Installation der Sensoren**

#### **3.3.1 Allgemeine Handhabungshinweise der Sensoren**

Der Radarsensor ist eine integrierte Einheit zur Messung von Entfernung und radialer Relativgeschwindigkeit. Vor der Montage des Sensors an dem Fahrzeug sind keine Änderungen, wie z.B. Voreinstellung oder Ausbau der Justierschrauben, Demontage des Sensors von der Halterung, Entfernung der Haltefedern erlaubt. [31]

Um Probleme beim Ausrichten zu vermeiden, ist eine Verschmutzung des Ausrichtespiegels (am Radom) zu vermeiden. Der Sensor ist mit einer Druckausgleichsmembran unter dem Typenschild ausgestattet. Beschädigungen können die Wasserdichtheit des Sensors beeinträchtigen. Beschädigte Teile müssen daher verschrottet werden.

Die Kontaktstifte dürfen nicht berührt werden, um Probleme durch elektrostatische Entladung zu vermeiden. Es wird auch empfohlen, den Kabelbaumstecker ohne angelegte Spannung anzuschließen, um die Erzeugung von Fehlereinträge zu vermeiden.

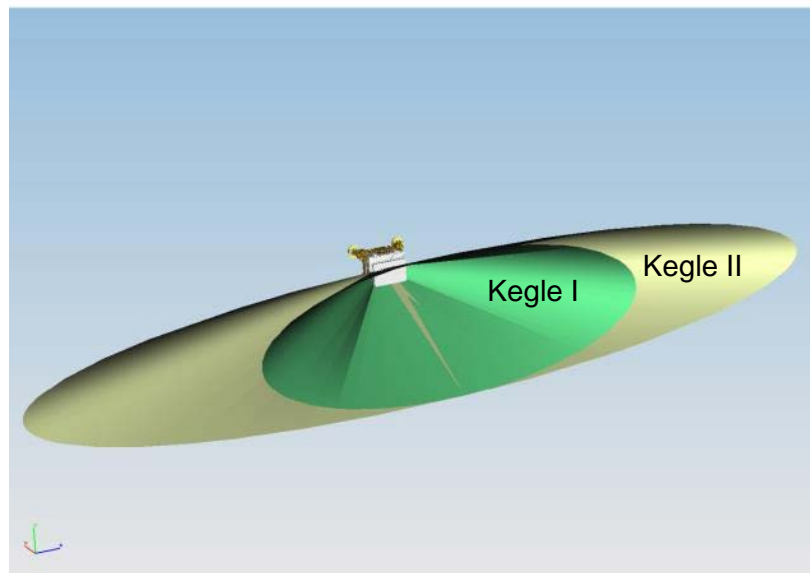
Zur Montage des Kabelbaumsteckers dürfen keine ungeeigneten Werkzeuge, wie z.B. Schraubendreher verwendet werden. Der ordnungsgemäße Eingriff des Kabelbaumsteckers muss überprüft werden. Bei allen Montagevorgängen muss besondere Aufmerksamkeit eingehalten werden, um mechanische Belastungen des Sensors zu vermeiden, die Beschädigungen verursachen können, wie z.B. das Herausspringen der Ausrichtungsschrauben von den Klammern aufgrund hoher Kräfte, die während des Einsteckens des Kabelbaumsteckers aufgebracht werden. Der Kabelbaum muss mindestens in einem Abstand von weniger als 20 cm an der Halterung befestigt werden.

#### **3.3.2 Installationsanleitung**

Der Sensor ist im vorderen Bereich des Fahrzeugs im Stoßfänger- oder Kühlergrillbereich installiert. Das Radom des Sensors zeigt in Fahrzeugfahrtrichtung und der Stecker zeigt nach unten. Idealerweise wird der Sensor ohne zusätzliche Abdeckung oder einen lackierten Stoßfänger vor der Radarkuppel in das Fahrzeug integriert. Wenn der Sensor hinter einer Abdeckung, z.B. einem Stoßfänger oder einem Radom, installiert werden soll, muss besondere Sorgfalt der Auswahl des Abdeckmaterials, der Abdeckungsform, der Lackierung und der Position der Abdeckung relativ zum Sensor gelten. [31]



Der Bereich vor dem Radarsensor muss innerhalb des relevanten Antennenbereichs frei von jeglichen Teilen, mit der Ausnahme von Radommaterial im Falle einer verdeckten Installation. Dabei muss zwischen Metall (einschließlich Chromapplikationen und lackierten Strukturen) und Kunststoffteilen unterschieden werden. Der relevante Bereich für die beiden Materialgruppen kann durch zwei unterschiedliche "Radarkegel" ausgehend von einer definierten Grundfläche auf der Oberseite des Radarsensors beschrieben werden. Kegel I ( $\pm 45^\circ$ ) ist für Kunststoffteile relevant, Kegel II ( $\pm 65^\circ$ ) ist für metallische oder lackierte Teile relevant (siehe Abbildung 3-5). Wenn Masken verwendet werden, muss, um den Luftstrom für die Heizung sicherzustellen, verifiziert werden, dass der Abstand zur Oberfläche des Sensorradoms mindestens 5 mm beträgt.

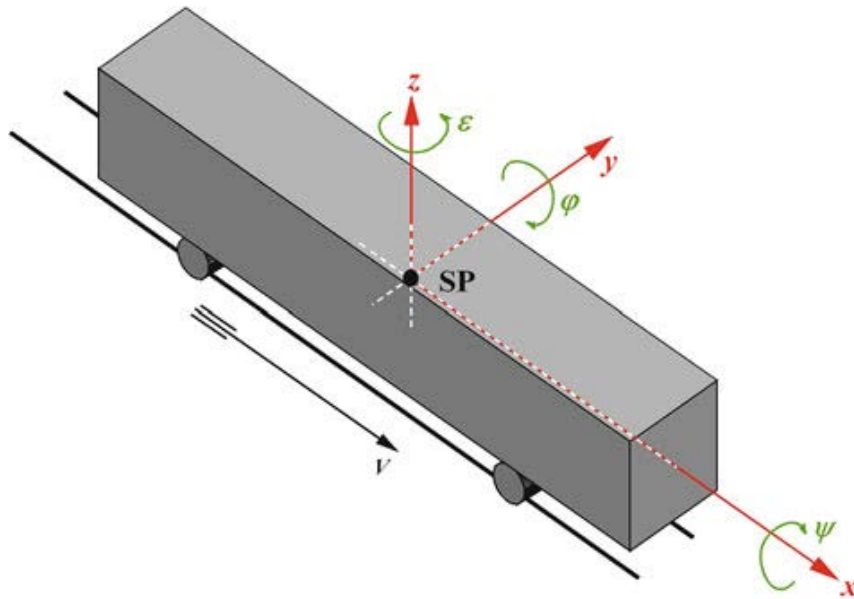


**Abbildung 3-5: Radarkegel der relevanten Bereiche [31]**

Die Radarkegel beschreiben einen "sicheren Bereich", in dem keine detaillierten Untersuchungen hinsichtlich der verzerrenden Wechselwirkung von Fahrzeugumgebung und Antennendiagramm erforderlich ist. In solchen Fällen, in denen Materialien die definierten Radarkegel in einem Bereich verletzen, bedeutet dies dennoch nicht, dass die Montageposition nicht akzeptabel ist. In diesem Fall müssen weitere detaillierte Untersuchungen durchgeführt werden, um den Montageort zu überprüfen.

### 3.3.3 Grundbegriffe zur Installationsvarianten

Zur Beschreibung der räumlichen Position eines Fahrzeugs dient das **Fahrzeugkoordinatensystem**, welches ein dreidimensionales kartesisches Koordinatensystem ist, in dem das Fahrzeug ruht. Handelt es sich um ein rechtshändiges Koordinatensystem, verläuft die  $x$ -Achse ausgehend vom Ursprung, wie in der Abbildung 3-6 dargestellt, auf der vertikalen Karosserieebene, waagerecht mit positiver  $x$ -Achse in Fahrtrichtung. [32]



**Abbildung 3-6: Fahrzeugkoordinatensystem [33]**

Um die Installationsvarianten der Sensoren zu erklären, werden zwei Begriffe **Azimut** und **Elevation** vorgestellt, die die Winkel der Sensorblickrichtung bezüglich des Fahrzeugkoordinatensystems beschreiben. (siehe Abbildung 3-7)

Der Azimutwinkel beschreibt die horizontale Ausrichtung der Sensorblickrichtung bezüglich  $x$ - $z$ -Ebene des Fahrzeugkoordinatensystems nach links (positiv) oder nach rechts (negativ). Im Vergleich dazu gibt der Elevationswinkel den vertikalen Winkel zwischen Sensorblickrichtung und die  $x$ - $y$ -Ebene an. Durch einen negativen Wert wird gekennzeichnet, dass die Sensorblickrichtung relativ nach unten zeigt.

Dies bedeutet beispielsweise, dass die Sensorblickrichtung genau in der Fahrtrichtung ist, wenn der Azimutwinkel und der Elevationswinkel beide null sind.

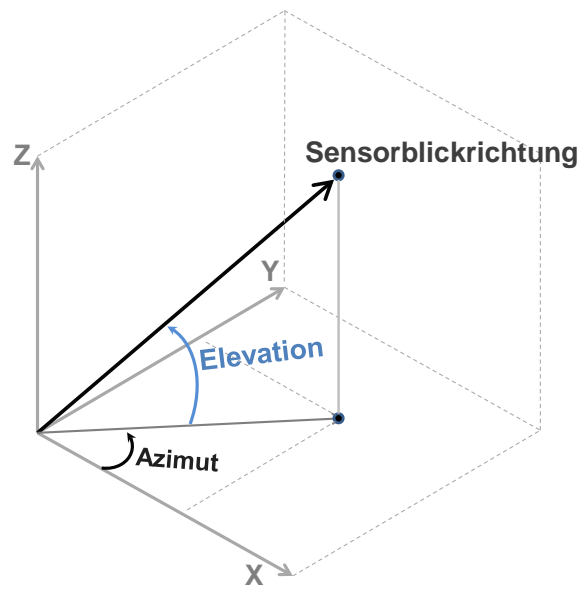


Abbildung 3-7: Azimut- und Elevationswinkel der Sensorblickrichtung

### 3.3.4 Installationsvarianten

Ein Benennungsverfahren in Format „AziXXEleYY“ wird verwendet um die Installationsvarianten in kompakter Form zu identifizieren und auch um die Dateien zu ordnen, wobei „Azi“ für „Azimutwinkel“ steht und „XX“ der Wert des Winkels in Grad ist.

Hauptsächlich sind drei Installationsvarianten zu implementieren, die jeweils „Azi0Ele0“ (siehe Abbildung 3-8), „Azi45Ele0“ (siehe Abbildung 3-9) und „Azi135Ele0“ (siehe Abbildung 3-10) heißen. Außerdem wird „Azi0Ele-45“ (siehe Abbildung 3-11) als Zusatzvariante implementiert.

Abbildung 3-8 zeigt die Variante „Azi0Ele0“, wobei der Radarsensor in Fahrzeugmitte mit Sicht genau in der Fahrzeugfahrtrichtung montiert wird. Durch den roten Kreis wird die Position bezüglich des Fahrzeuges gekennzeichnet.



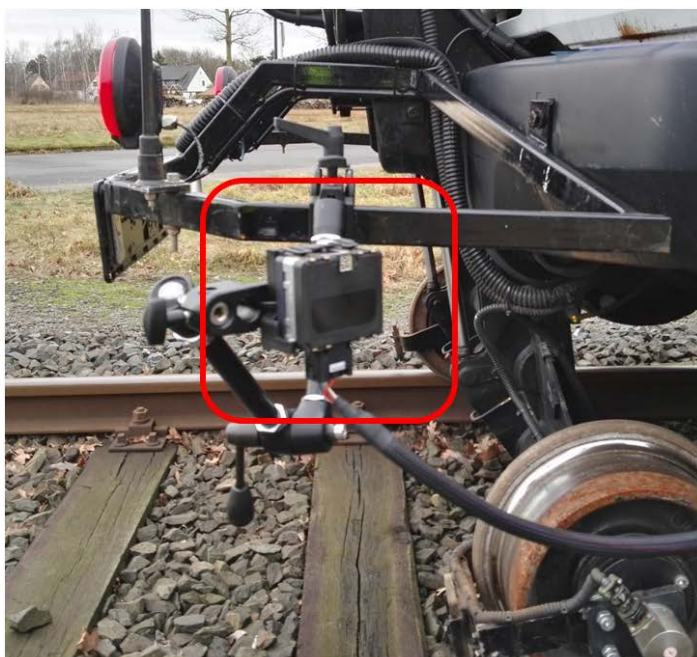
**Abbildung 3-8: Azi0Ele0 (Sensorblickrichtung in Fahrtrichtung)**



Abbildung 3-9 und Abbildung 3-10 zeigen jeweils die Variante „Azi45Ele0“ und „Azi135Ele0“, wobei die Sensorblickrichtung entsprechend schräg nach vorne und schräg nach hinten zeigt. Der Sensor wird durch einen Gelenkarm an der linken Seite des Stoßbalkens montiert.



**Abbildung 3-9: Azi45Ele0 (Sensorblickrichtung schräg nach vorne)**



**Abbildung 3-10: Azi135Ele0 (Sensorblickrichtung schräg nach hinten)**

Da die meisten typischen im Bahnbereich angewendeten Radarsensoren bzw. optischen Sensoren in der Lage sind, die Fahrzeuggeschwindigkeit relative zum Untergrund direkt zu messen, ist es auch wichtig zu überprüfen, ob der eingesetzte Automotive-Radarsensor diese Fähigkeit hat. Deswegen wird die Zusatzvariante „Azi0Ele-45“ mit Sensorblickrichtung schräg nach unten eingebaut, wie in der Abbildung 3-11 zu sehen. Dies geschieht um zu verifizieren, ob der Sensor den Untergrund als Objekt erkennen kann.



**Abbildung 3-11: Azi0Ele-45 (Sensorblickrichtung schräg nach unten)**

### 3.4 Durchführung der Messfahrten

#### Testfeld Hafenbahn Braunschweig

Neben der öffentlichen Eisenbahninfrastruktur (Serviceeinrichtung) steht der Hafenbahn Braunschweig ein eigener Rangierbetrieb mit Lokomotiven, Wagen und eigenes Betriebspersonal zur Verfügung. Auf der Hafenbahn finden ausschließlich Rangierfahrten durch Bedienstete der Anschlussbahn statt, sowie das den Anschluss bedienenden Eisenbahnverkehrs-Unternehmen (EVU), wie z. B. DB-Schenker-Rail. [34] Die Lageskizze der Hafenbahn bzw. die zum Zweck der Messfahrten befahrenen Gleise werden im Anhang A angefügt.

Gemäß der Lageskizze befindet sich zwischen km 1,6 und km 2,3 der dreigleisige Übergabebahnhof mit den Handweichen 3 bis 7. Die Nutzlänge von Gleis 1 bis 3, auf dem das Testfahrzeug RailDriVE® Messfahrten durchführen kann, beträgt jeweils 546 m, 519 m und 550 m. Alle Fahrzeugbewegungen auf dem Zuführgleis und im Übergabebahnhof sind Rangierfahrten und finden auf Sicht statt. Das Abstoßen von Waggonen ist verboten. Die zugelassene Höchstgeschwindigkeit für Rangierbetrieb beträgt 20 km/h. [34]

Um die Zustandsüberwachung zu erproben, hat das DLR seit ein paar Jahren mit der Hafengesellschaft Braunschweig zusammengearbeitet. Die Hafengesellschaft betreibt im Braunschweiger Binnenhafen zwei Rangierlokomotiven auf dem über 15 km langes Schienennetz, welche auch die Anbindung an das nationale Hauptnetz besitzt. Außerdem betreibt DLR in Braunschweig eine Anwendungsplattform für Intelligente Mobilität (AIM), zu der das Hafengebiet auch gehört. Seit 2015 rüstet das DLR Multisensorsysteme für die Rangierlokomotiven aus und zeichnet kontinuierlich Messdaten für die Entwicklung und Erprobung von Georeferenz- und Zustandsüberwachungsverfahren auf. [35]

Nach der Vereinbarung mit der Braunschweig Hafenbetriebsgesellschaft kann die Strecke zum Testen befahren werden. Eine der Messfahrten wird in Abbildung 3-12 anschaulich dargestellt.





Abbildung 3-12: Route der Messfahrten



## 4 Algorithmen zur Datenverarbeitung

Die Ausgabedaten vom MRR-Sensor werden je nach Verarbeitungszeit ungefähr alle 40 bis 70 Millisekunden als ein großer Datenblock (als „Cycle“ genannt) gesendet. In jedem Cycle können maximal 32 Objekte mit jeweiliger CAN-ID erkannt werden. Die wichtigsten Daten jedes Objektes, z.B. radialer Abstand (`obj_dr`), radiale Geschwindigkeit (`obj_vr`) und Azimutwinkel (`obj_phi`) sind in den entsprechenden 8-Bytes CAN-Daten gespeichert. Falls zu einer CAN-ID kein Objekt erkannt wird, werden alle 64-bits der CAN-Daten durch Nullen besetzt.

Laut der im Datenblatt beschriebenen Struktur der CAN-Daten können hierfür die Werte von `obj_dr`, `obj_vr` und `obj_phi` mit dem im Kapitel 3.2 dargestellten Verfahren für Luxact CAN-Messages dekodiert bzw. ausgerechnet werden.

### 4.1 Umrechnung der Radialgeschwindigkeiten

Da auf Basis des Doppler-Effektes nur die radiale Relativgeschwindigkeit gemessen wird, welche als Variable `obj_vr` gekennzeichnet wird, wird der Zusammenhang zwischen der radialen Relativgeschwindigkeit und der Relativgeschwindigkeit in Fahrzeughfahrtrichtung durch Gleichung (4.1) beschrieben.

$$v_{\text{radial}} = v_x \cdot \cos \alpha \quad (4.1).$$

Dabei bezeichnet  $\alpha$  ist den Azimutwinkel des Objektes bezüglich der Fahrzeughfahrtrichtung (siehe Abbildung 4-1).

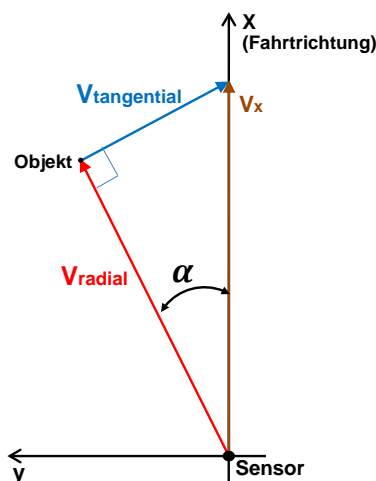


Abbildung 4-1: Radial- und Tangentialkomponenten in Bezug auf Dopplereffekt

Bezüglich Winkel  $\alpha$  gilt jedoch allgemein:

$$\alpha = Azi\_sensor + obj\_phi \quad (4.2).$$

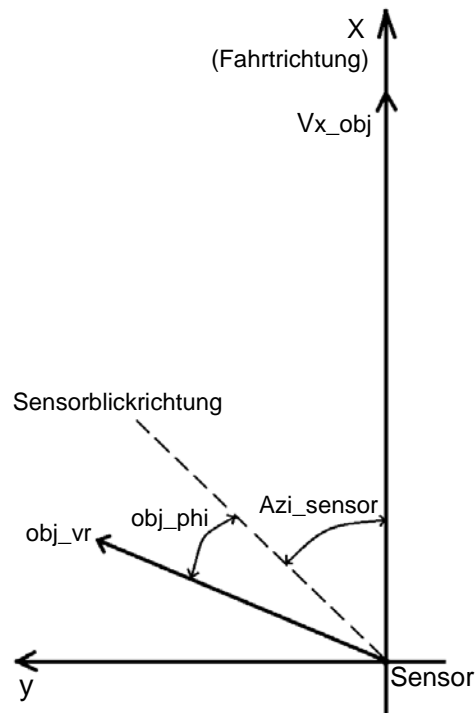
wobei,

*Azi\_sensor*: Azimutwinkel der Sensorblickrichtung bezüglich der Fahrzeugfahrtrichtung

*obj\_phi*: Azimutwinkel des Objektes bezüglich der Sensorblickrichtung

Die umzurechnende Relativgeschwindigkeit des erkannten Objektes in der Fahrzeugfahrtrichtung  $v_{x\_obj}$  ergibt sich durch die Kombination von (4.1) und (4.2)

$$v_{x\_obj} = \frac{obj\_vr}{\cos(Azi\_sensor + obj\_phi)} \quad (4.3).$$



**Abbildung 4-2: Umrechnung der relativen Radialgeschwindigkeit**

Für die drei Installationsvarianten „Sensorblickrichtung in Fahrtrichtung“, „Sensorblickrichtung schräg nach vorne“ und „Sensorblickrichtung schräg nach hinten“ beträgt *Azi\_sensor* in der Abbildung 4-2 jeweils 0 Grad, 45 Grad und 135 Grad.

## 4.2 Bestimmung der Fahrzeuggeschwindigkeit

Obwohl der eingesetzte MRR-Sensor die relativen Radialgeschwindigkeiten der Objekte zum Sensor ausgeben kann, ist dieser Sensor nicht in der Lage, den Bewegungszustand von jedem erkannten Objekt zu bestimmen. Allerdings ist die umgerechnete Relativgeschwindigkeit  $v_{x\_obj}$  nur, wenn das Objekt sich im Stillstand befindet, gleich der Fahrzeuggeschwindigkeit. Um damit die Fahrzeuggeschwindigkeit zu berechnen, müssen alle solche sogenannten benutzbaren  $v_{x\_obj}$  gefiltert werden.

Unter Berücksichtigung der Beschleunigungsfähigkeit des Fahrzeuges (z.B. einer Rangierlokomotive), auf das der Sensor installiert wird, gilt folgende Bedingung für die Differenz der Fahrzeuggeschwindigkeit von zwei aufeinanderfolgende Cycles.

$$|v_{Fzg\_Cycle_n} - v_{Fzg\_Cycle_{n+1}}| \leq a_{max} \cdot \Delta t_{max} \quad (4.4).$$

Wenn die Differenz von der vorherigen Fahrzeuggeschwindigkeit und der umgerechneten Relativgeschwindigkeit  $v_{x\_obj}$  des Objekts kleiner als dieser Schwellwert ist, wird diese  $v_{x\_obj}$  als benutzbar betrachtet.

$$|vr_{previous} - v_{x\_obj\_benutzbar}| \leq v_{Schwellwert} \quad (4.5).$$

Dabei ist  $v_{Schwellwert} = a_{max} \cdot \Delta t_{max}$  ein für jeglichen Fahrzeugtyp unter einer bestimmten Betriebsart (z.B. Rangierbetrieb) zu kalibrierender Schwellwert. In der Abbildung 4-3 wird das Diagramm des Algorithmus dargestellt.

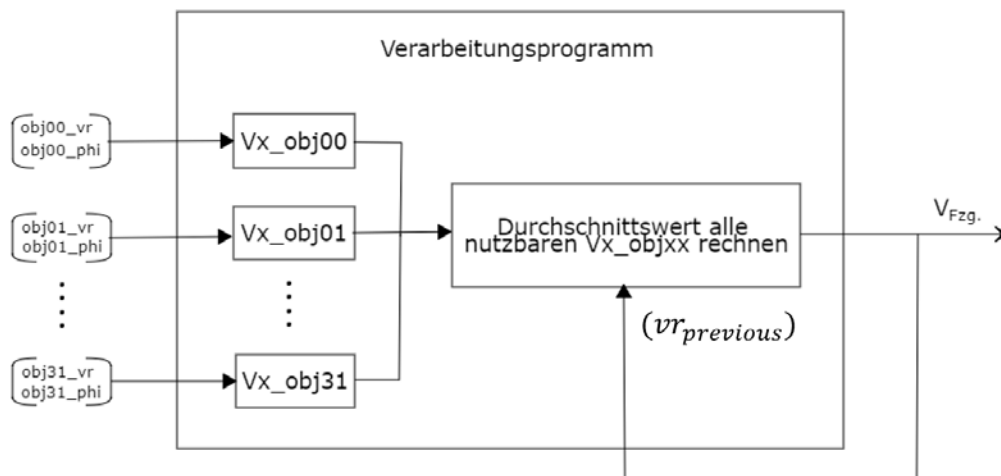
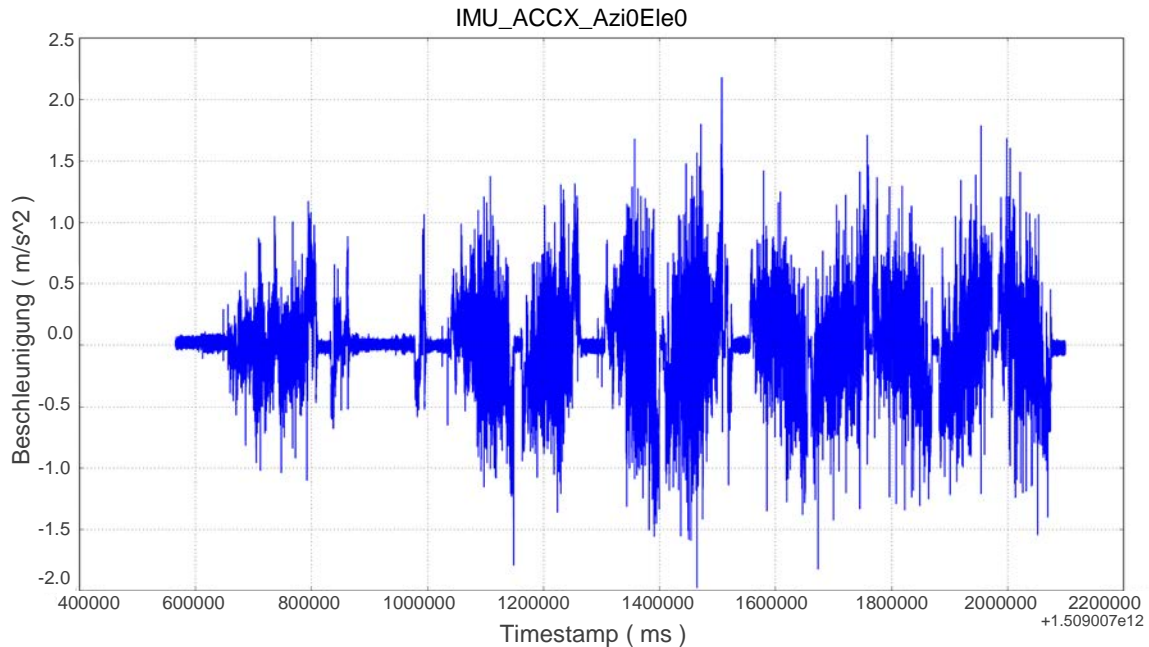


Abbildung 4-3: Algorithmus zum Rechnen der Fahrzeuggeschwindigkeit

Um diesen Schwellwert vom Messfahrzeug RailDrIVE zu bestimmen, werden die Messwerte einer auf das Fahrzeug installierten inertialen Messeinheit betrachtet. Abbildung 4-4 zeigt die Beschleunigungswerte des Messfahrzeuges in x-Achse.



**Abbildung 4-4: IMU-Werte in x-Achse (Fahrzeugfahrtrichtung)**

Der maximale absolute Wert beträgt  $2.1788 \frac{m}{s^2}$  und 80 ms wird als das maximale Zeitintervall zwischen zwei aufeinanderfolgenden Cycles des MRR-Sensors angenommen.

$$v_{Schwellwert} = \max(|IMU_{ACCX}|) \cdot \max(\Delta t_{cycle}) = 2.1788 \frac{m}{s^2} \cdot 0.08s = 0.174 \frac{m}{s} \quad (4.6)$$

D.h. für RailDrIVE wird der Schwellwert von  $0.174 \frac{m}{s}$  zur Filterung im Programm verwendet. Um anschließend auf Basis der benutzbaren umgerechneten Relativgeschwindigkeit  $v_{x,obj}$  die Fahrzeuggeschwindigkeit zu berechnen, kann entweder das arithmetische Mittel

$$V_{Fzg.} = \frac{\sum V_{x,obj\_benutzbar}}{n_{x,obj\_benutzbar}} \quad (4.7)$$

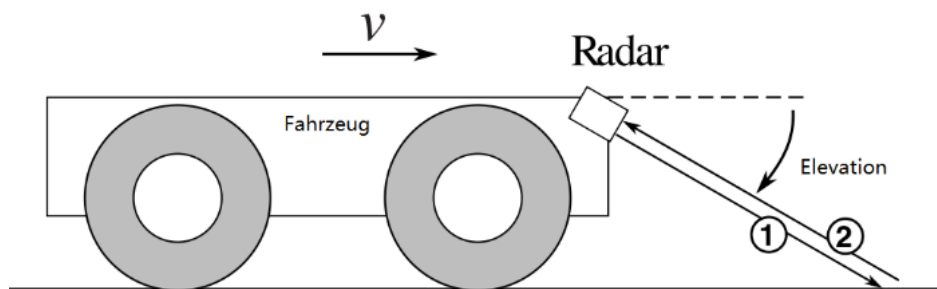
oder der Median angewendet werden. Die Programmcodes zum Algorithmus werden im Anhang B angefügt.

## 5 Messdatenverarbeitung und Ergebnisdokumentation

In diesem Kapitel werden die in Kapitel 4 vorgestellten Algorithmen auf die Messdaten angewendet. Die mit Hilfe des MRR-Sensors abgeleiteten Geschwindigkeiten werden mit den Messwerten von anderen schon auf dem Messfahrzeug RailDrive zur Verfügung stehenden Sensoren verglichen. Da das JavadSigma (kurz „Javad“) ein hochwertiges GNSS-Gerät ist, werden die davon gemessenen Geschwindigkeiten als Referenz betrachtet. Obwohl es keinen bestimmten „Wert“ gibt um die Messgenauigkeit der Geschwindigkeit zu beschreiben, gibt es jedoch innerhalb der empfangenen NMEA-Messages die „GSA“ und „GST“ Messages, in welchen die DOPs (Dilution of Precision) und die Genauigkeit der Positionen angegeben sind. Die Standardabweichungen von latitudinalen und longitudinalen Fehler sind jeweils weniger als einen Meter. Darüber hinaus können die Messwerte zusätzlich mit einem anderem Doppler-Radarsensor (Deuta) und einem optischen Geschwindigkeitssensor (Luxact) verglichen werden.

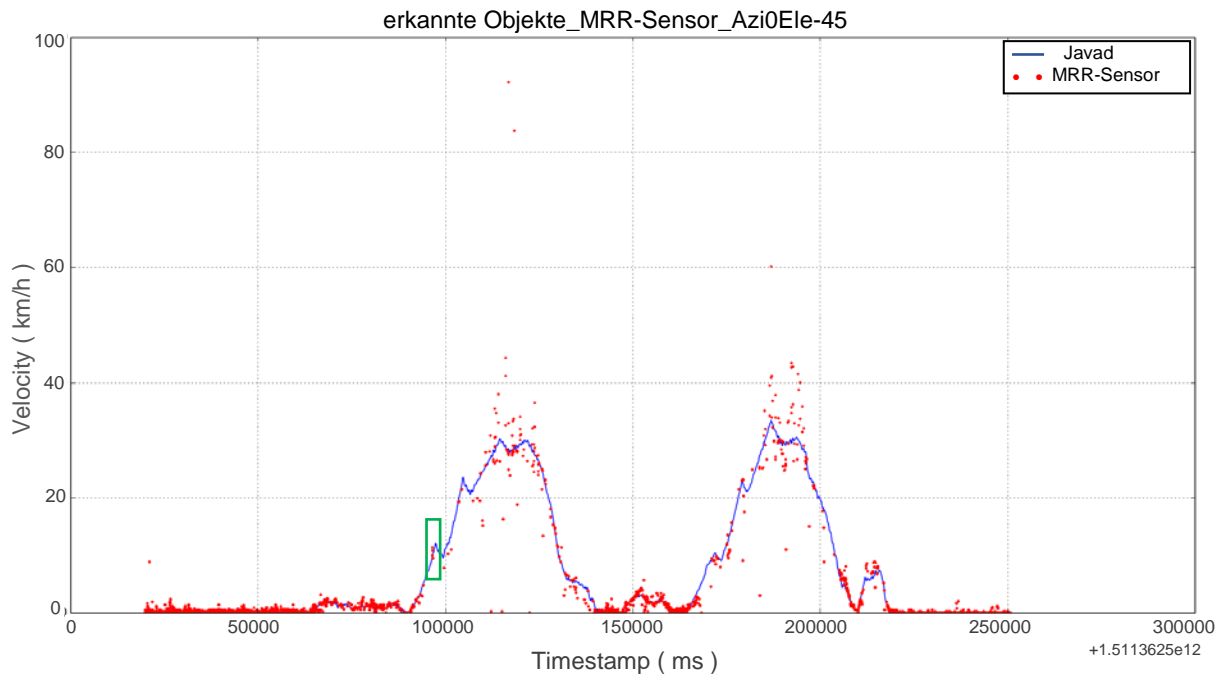
### 5.1 Sensorblickrichtung schräg nach unten

Um die Eigenschaft des MRR-Sensors, ob er den Untergrund als Objekt erkennt, zu überprüfen wird eine Messfahrt mit der Sensorblickrichtung schräg nach unten durchgeführt. (siehe Abbildung 5-1)



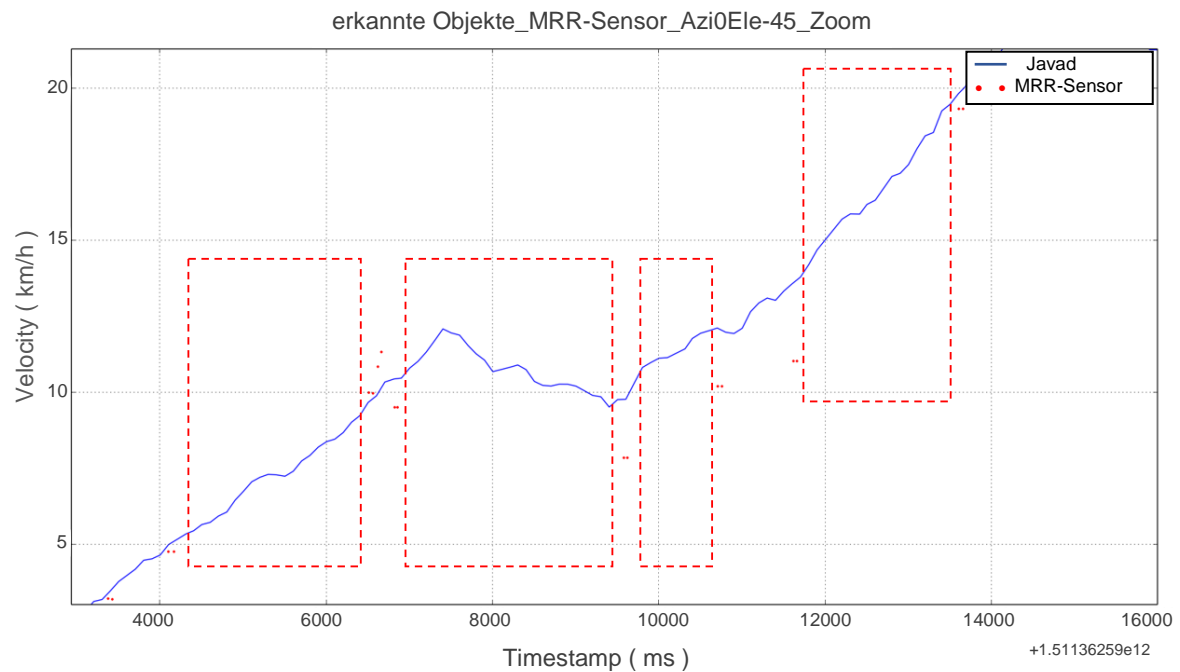
**Abbildung 5-1: Sensorblickrichtung schräg nach unten (vgl. [36])**

Die Messfahrt dauert ungefähr drei Minuten. In Abbildung 5-2 zeigt die blaue Linie den durch den GNSS-Sensor (JavadSigma) im Standalone-Modus gemessenen Geschwindigkeitsverlauf. Jeder rote Punkt steht für die durch den MRR-Sensor gemessene bzw. umgerechnete Geschwindigkeit eines Objektes.



**Abbildung 5-2: erkannte Objekte mit Sensorblickrichtung schräg nach unten**

Um ihn deutlicher zu sehen wird der im grünen Kasten markierte Teil der Abbildung 5-2 vergrößert (siehe Abbildung 5-3). Es ist deutlich zu sehen, dass der Sensor sehr oft gar kein Objekt erkennt, wie beispielsweise innerhalb der in den roten Kästen gekennzeichneten Zeitintervalle, obwohl der Untergrund sich stets im Sichtbereich des Sensors befindet.



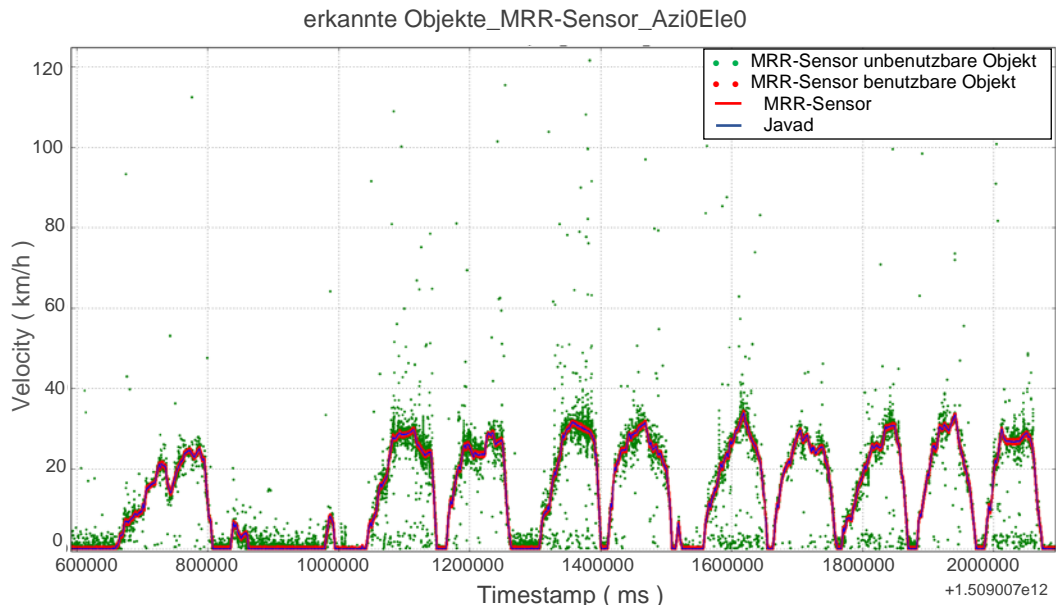
**Abbildung 5-3: vergrößerter Teil der Abbildung 5-2**

Somit wird eine Eigenschaft des MRR-Sensors verifiziert, dass der Untergrund nicht als Objekt durch den Sensor erkannt wird, weshalb es nicht möglich ist, aus der gemessenen Relativgeschwindigkeit des Sensors bezüglich des Untergrundes die Fahrzeuggeschwindigkeit direkt abzuleiten.

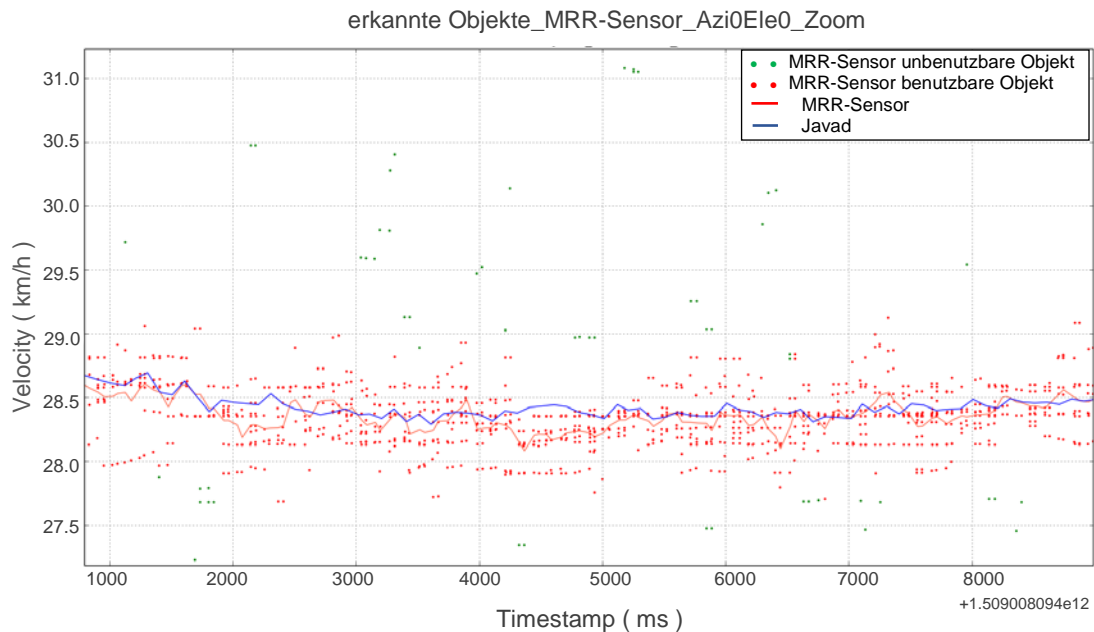
## 5.2 Sensorblickrichtung in Fahrtrichtung

Nachdem die Messfahrt mit Sensorblickrichtung in Fahrtrichtung durchgeführt wird, werden die vorgestellten Algorithmen auf die Messdaten angewendet.

In der Abbildung 5-4 werden die Objekte mit benutzbaren Relativgeschwindigkeiten als rote Punkte gekennzeichnet. Die grünen Punkte sind alle die Objekte mit unbenutzbaren Relativgeschwindigkeiten.



**Abbildung 5-4: erkannte Objekte mit Sensorblickrichtung in Fahrtrichtung**

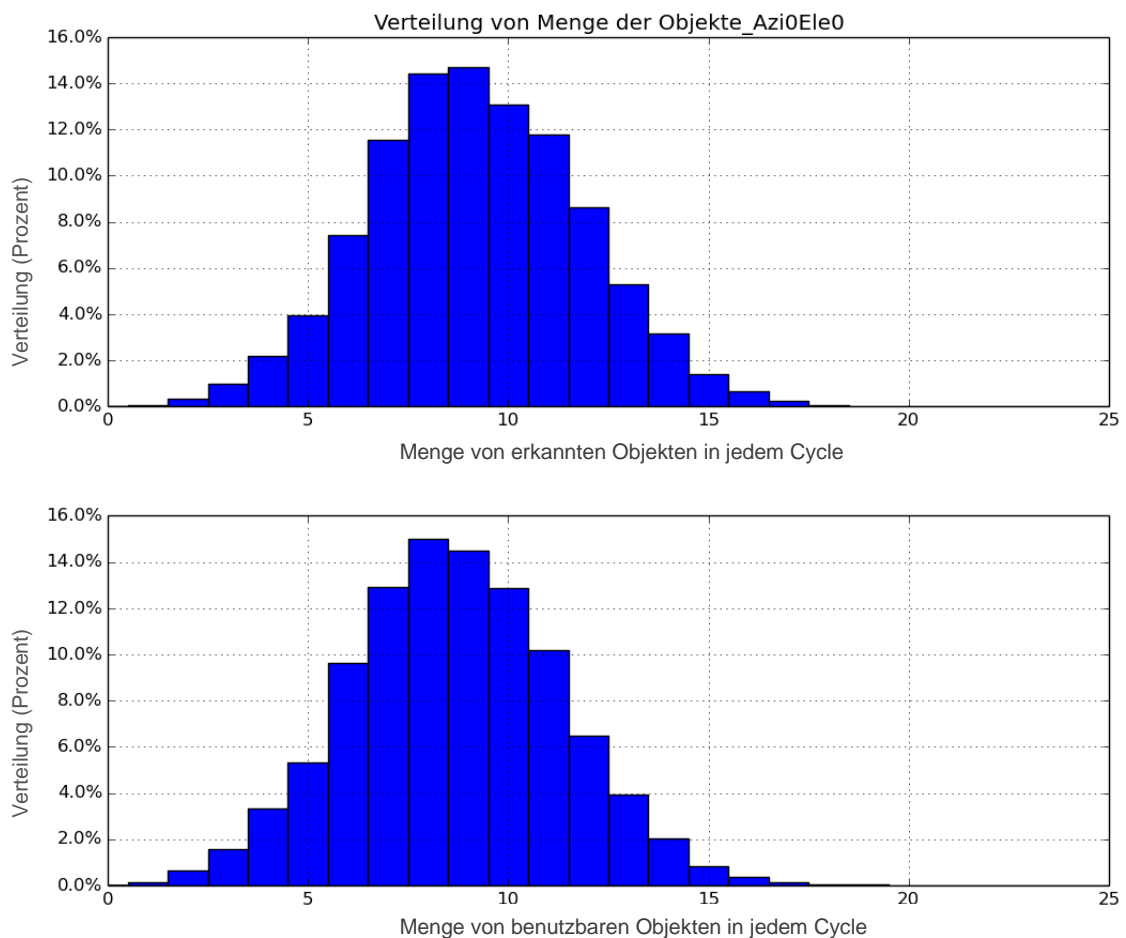


**Abbildung 5-5: vergrößerter Teil der Abb. 5-4**



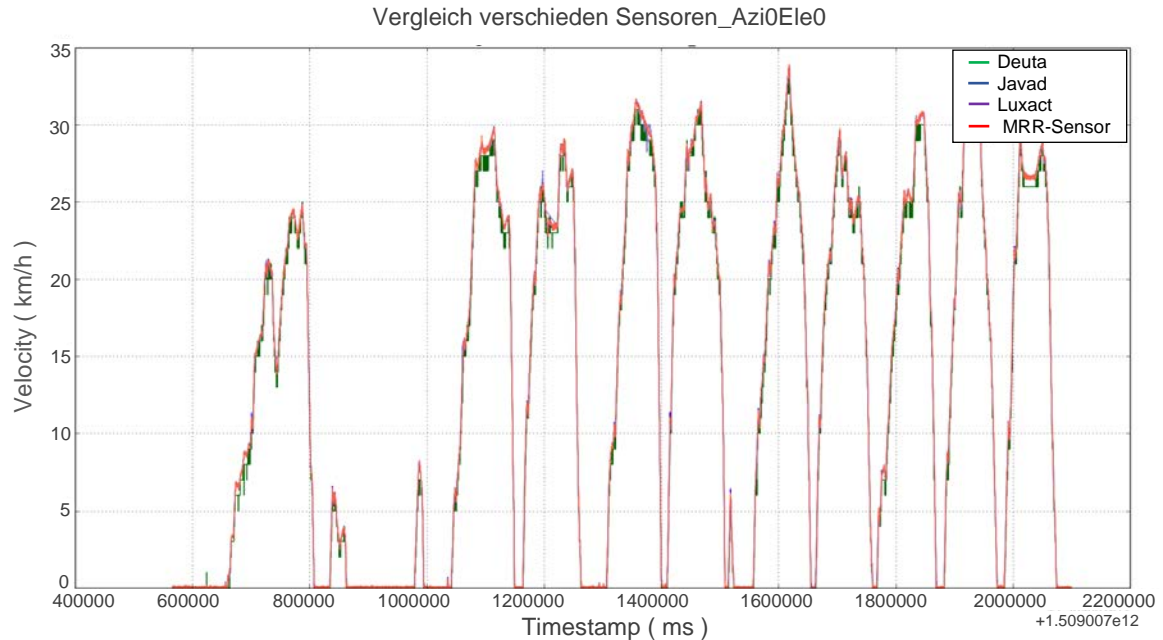
Wenn ein Teil des Diagramms vergrößert wird, sieht man in Abbildung 5-5, dass die benutzbaren Punkte sich der blauen Linie (gemessene Geschwindigkeit auf Basis des GNSS-Sensors) annähern. Die rote Linie zeigt dann die dadurch berechnete Fahrzeuggeschwindigkeit.

Um die Menge von allen erkannten Objekten und benutzbaren Objekten genau zu sehen werden die Histogramme in Abbildung 5-6 dargestellt. Es wird angezeigt, dass es in fast allen Cycles nicht nur genügende (mehr als vier) erkannte Objekte, sondern auch mindestens ein benutzbares Objekt gibt.

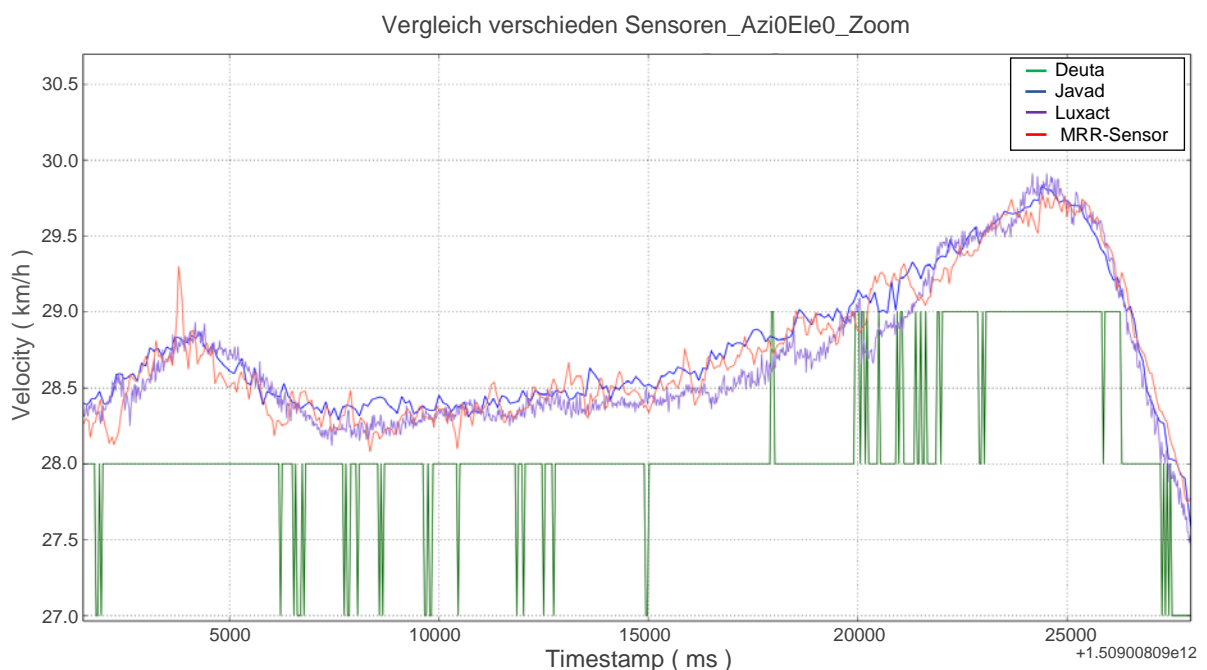


**Abbildung 5-6: Verteilung von Menge der Objekte mit Sensorblickrichtung in Fahr-  
richtung**

Damit die Ergebnisse anschaulich gezeigt werden können, werden die berechneten Fahrzeuggeschwindigkeiten zusammen mit den Messwerten aus anderen Sensoren in der Abbildung 5-7 bzw. teilweise vergrößert in Abbildung 5-8 in einem Bild dargestellt. Es ist zu sehen, dass die Verläufe gut miteinander übereinstimmen.

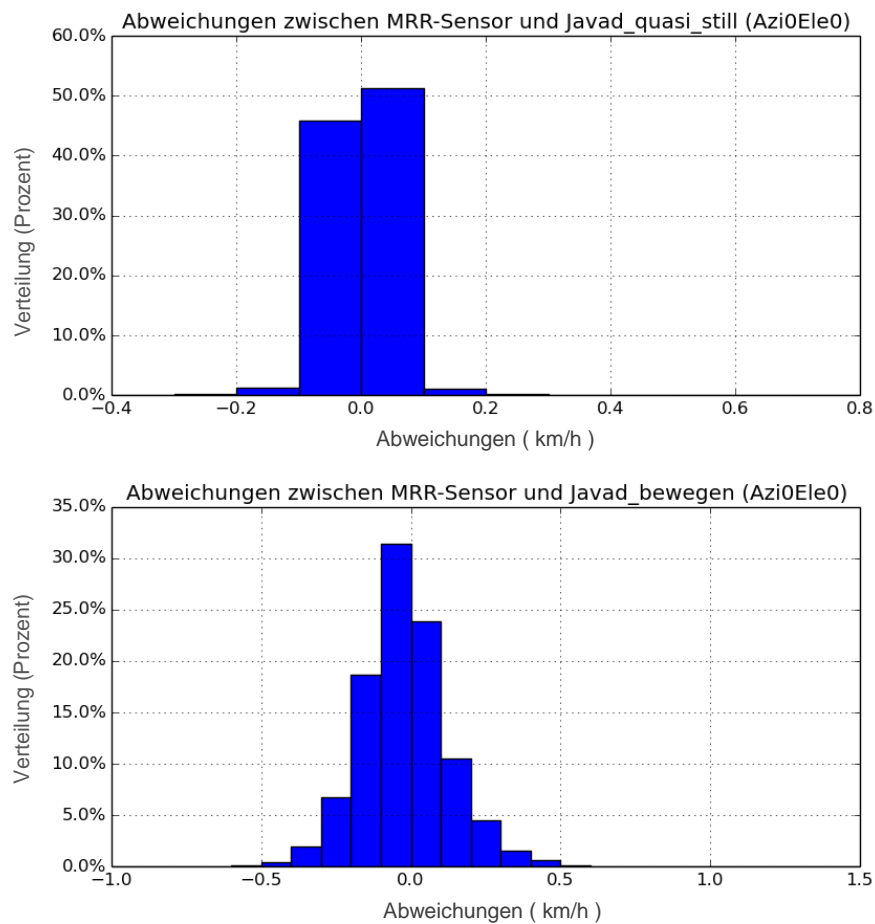


**Abbildung 5-7: Geschwindigkeitsverläufe von verschiedenen Sensoren**



**Abbildung 5-8: vergrößerter Teil der Geschwindigkeitsverläufe von verschiedenen Sensoren**

Darüber hinaus werden die Abweichungen der Messwerte zwischen dem MRR-Sensor und andere Sensoren statistisch bewertet um die Messqualität zu überprüfen. Die Abweichungen werden in zwei unterschiedlichen Fällen betrachtet, nämlich Fahrzeug im Quasistillstand und Fahrzeug in Bewegung (Fahrzeuggeschwindigkeit größer als 0,5 km/h). Wenn die Messwerte vom GNSS-Gerät (Javad) als Referenz verwendet werden, sind die Abweichungen dazu in Abbildung 5-9 und in Tabelle 5.1 angezeigt.



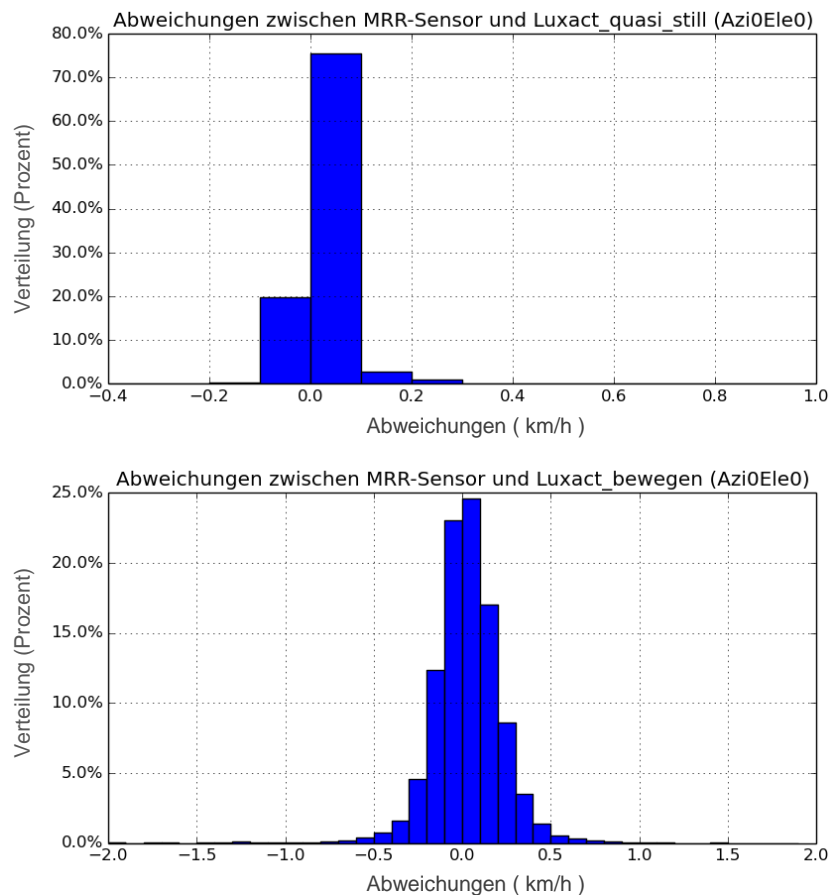
**Abbildung 5-9: Abweichungen der Geschwindigkeiten zwischen MRR-Sensor und GNSS-Sensor (Javad)**

**Tabelle 5.1: Abweichungen der Geschwindigkeiten zwischen MRR-Sensor und Javad**

	Maximum (km/h)	Minimum (km/h)	Standardabweichung (km/h)
<b>Quasistillstand</b>	0.657	-0.278	0.041
<b>Bewegung</b>	0.957	-0.830	0.144

Hier ist zu sehen, dass fast alle Abweichungen kleiner als 0,5 km/h sind und dass beim Quasistillstand die absoluten Fehler auch deutlich kleiner sind als wenn das Fahrzeug sich bewegt.

Zusätzlich werden die Abweichungen der Messwerte zwischen MRR-Sensor und den optischen Sensor (Luxact) in Abbildung 5-10 und in Tabelle 5.2 angezeigt. Die meisten Abweichungen sind ebenso kleiner als 0,5 km/h. Außerdem ist die Standardabweichung nur noch 0,207 km/h, wenn das Fahrzeug sich bewegt, wobei sich eine gute Messqualität ergibt.



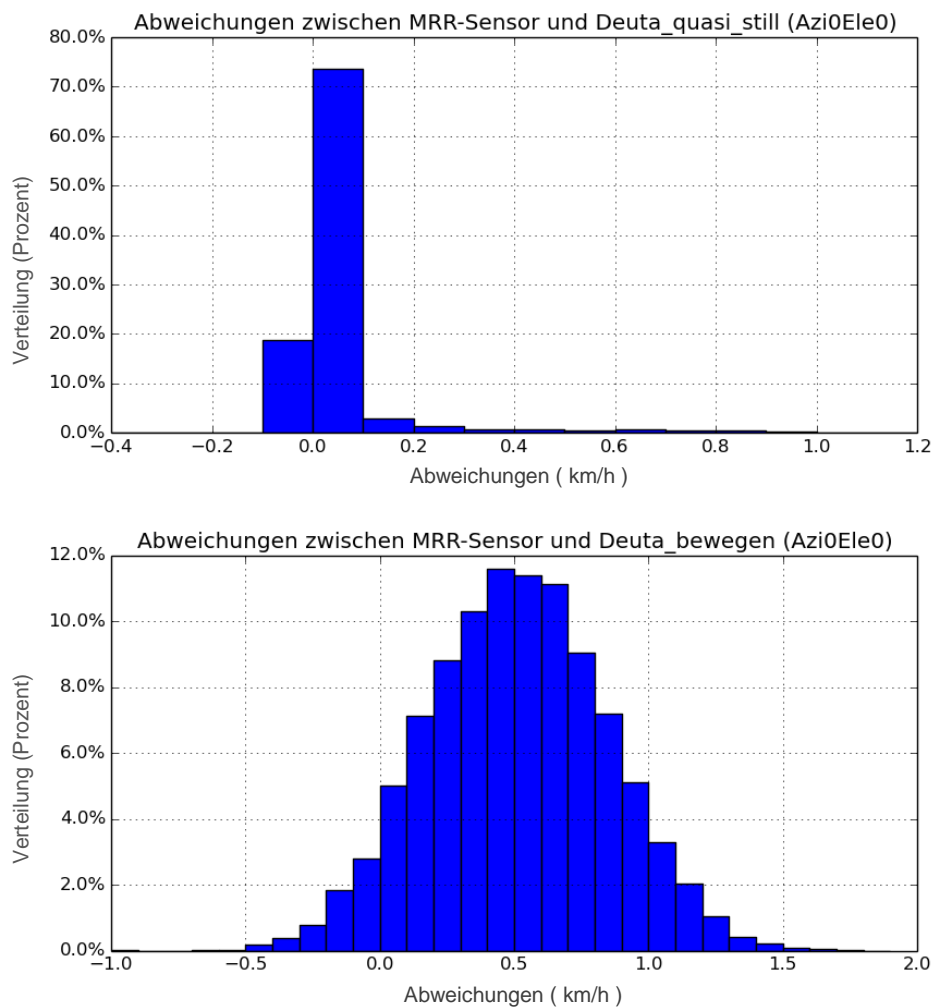
**Abbildung 5-10: Abweichungen der Geschwindigkeiten zwischen MRR-Sensor und Optisch-inertiale-Sensor (Luxact)**

**Tabelle 5.2: Abweichungen der Geschwindigkeiten zwischen MRR-Sensor und Luxact**

	Maximum (km/h)	Minimum (km/h)	Standardabweichung (km/h)
<b>Quasistillstand</b>	0.815	-0.344	0.051
<b>Bewegung</b>	1.614	-2.088	0.207

Gleichermaßen zeigen Abbildung 5-11 und Tabelle 5.3 die Abweichungen zwischen MRR-Sensor und einem anderen Doppler-Radar (Deuta).

Es ist zu sehen, dass die Abweichungen zwischen MRR-Sensor und Deuta deutlich größer als bei den anderen zwei Sensoren sind. Außerdem ist aus der Abbildung 5-11 zu erkennen, dass zwischen den Messwerten vom MRR-Sensor und Deuta ein systematischer Fehler von ca. 0,5 km/h existiert, welcher der Verschiebung von Geschwindigkeitsverläufen in der Abbildung 5-8 entspricht.



**Abbildung 5-11: Abweichungen der Geschwindigkeiten zwischen MRR-Sensor und Doppler-Radar (Deuta)**

**Tabelle 5.3: Abweichungen der Geschwindigkeiten zwischen MRR-Sensor und Deuta**

	<b>Maximum (km/h)</b>	<b>Minimum (km/h)</b>	<b>Standardabweichung (km/h)</b>
<b>Quasistillstand</b>	1.084	-0.285	0.119
<b>Bewegung</b>	2.454	-0.960	0.336

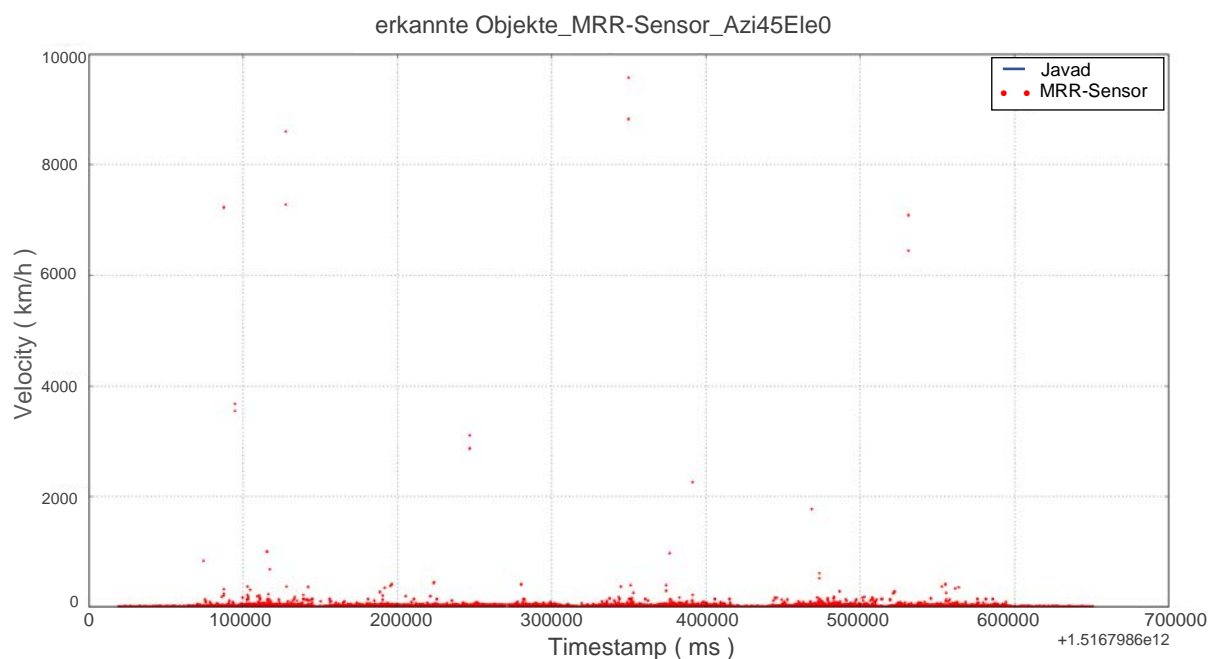
Auf Basis der Vergleiche zeigt sich, dass die Messqualität des MRR-Sensors mit Sensorblickrichtung in Fahrtrichtung mit dem GNSS-Sensor (im Standalone-Modus) bis auf relative kleine Abweichungen übereinstimmt.

### 5.3 Sensorblickrichtung schräg nach vorne und hinten

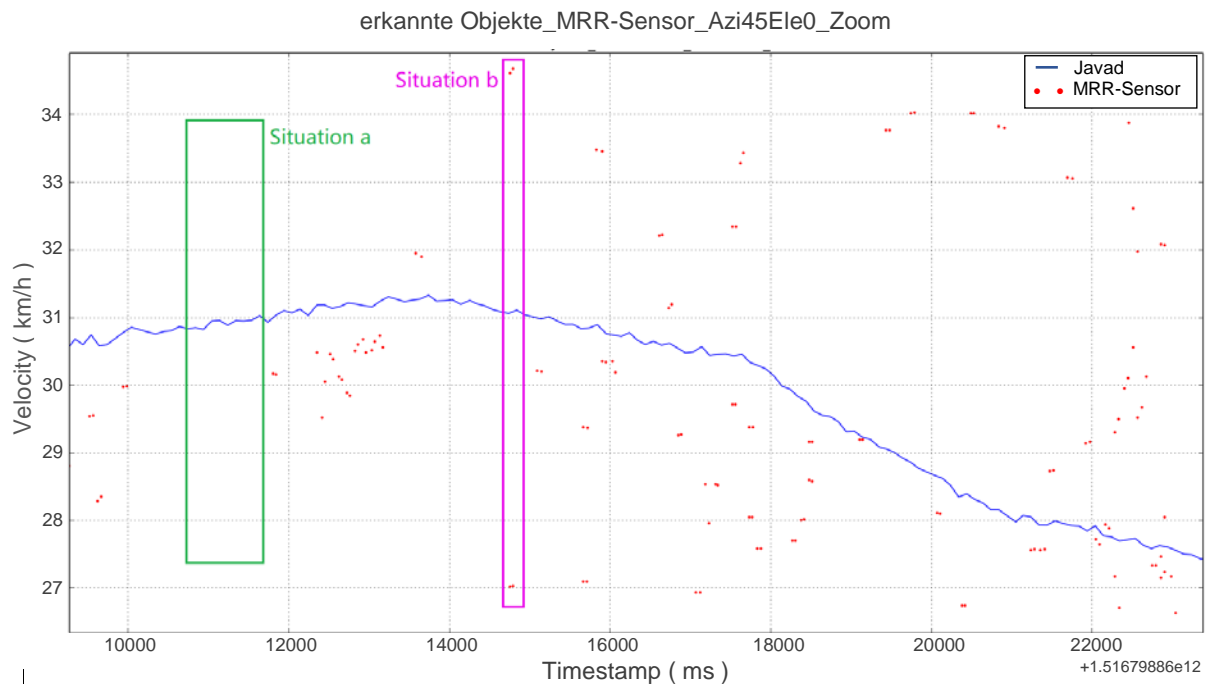
Wenn der MRR-Sensor auf dem Messfahrzeug so installiert wird, sodass die Sensorblickrichtung schräg nach vorne oder hinten zeigt, dann ist dieser nicht in der Lage, die entwickelten Bearbeitungsalgorithmen auf die Messdaten anzuwenden, da sich oft bei solchen Installationsvarianten die Mehrheit der umgerechneten Relativgeschwindigkeiten der Objekte in einer der folgenden zwei Situationen befinden.

- Es gibt keine umgerechnete Relativgeschwindigkeit bzw. kein erkanntes Objekt in einem Cycle.
- Die Differenzen von allen umgerechneten Relativgeschwindigkeiten bezüglich den entsprechenden Messwerte aus Javad sind zu groß.

Es werden alle umgerechneten Relativgeschwindigkeiten der Objekte in der Abbildung 5-12 anschaulich als rote Punkte dargestellt. Zudem werden die zwei Situationen in der teilweise vergrößerten Abbildung 5-13 dargestellt.



**Abbildung 5-12: erkannte Objekte mit Sensorblickrichtung schräg nach vorne**

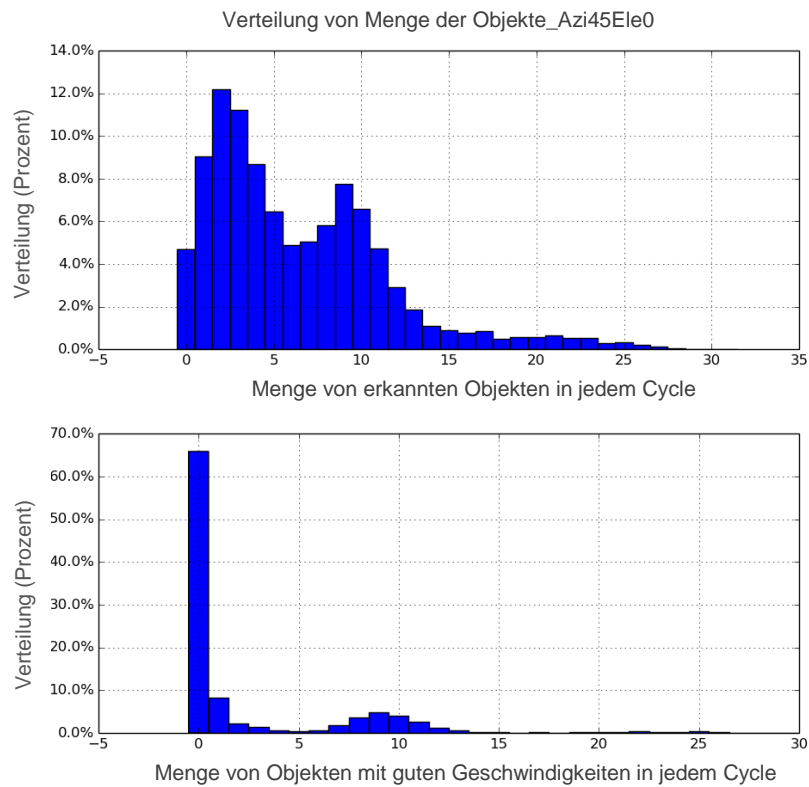


**Abbildung 5-13: vergrößerter Teil der Abbildung 5-12 mit den zwei Situationen**

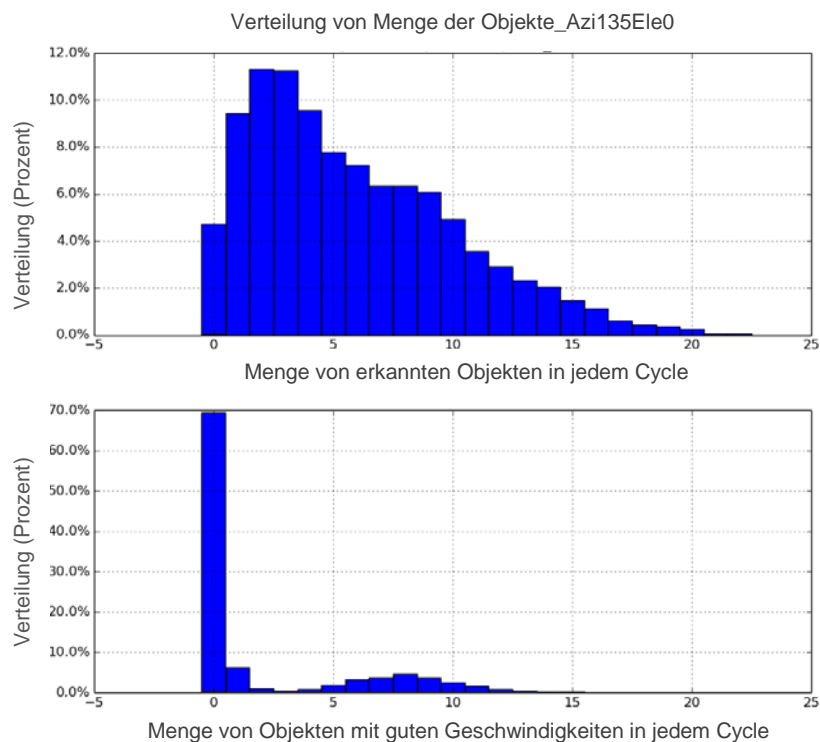
Auch um die Verteilungen der umgerechneten Geschwindigkeiten statistisch zu bewerten wird eine Toleranz von  $\pm 1 \text{ km/h}$  bezüglich des entsprechenden Messwertes von Javad definiert. Die umgerechneten Relativgeschwindigkeiten innerhalb der Toleranz werden nachstehend als „gute Geschwindigkeiten“ bezeichnet.

Die Menge von erkannten Objekten in jedem Cycle und die Menge von Objekten mit guten Geschwindigkeiten in jedem Cycle bei Sensorblickrichtung schräg nach vorne und hinten werden jeweils in Abbildung 5-14 und Abbildung 5-15 gezeigt. Im Gegensatz zum Fall, wo die Sensorblickrichtung in Fahrtrichtung zeigt (siehe Abbildung 5-6), ist noch deutlicher zu sehen, dass die Menge der gesamt erkannten Objekte relative weniger sind. Außerdem wird in mehr als 65% allen Messcycles kein Objekt mit guter Relativgeschwindigkeit erkannt. Da es hier nämlich keine benutzbaren Eingabewerte für den Algorithmus gibt, kann die Fahrzeuggeschwindigkeit deswegen nicht bestimmt werden wenn die Sensorblickrichtung schräg nach vorne oder hinten zeigt.





**Abbildung 5-14: Verteilung von Menge der Objekte mit Sensorblickrichtung schräg nach vorne**



**Abbildung 5-15: Verteilung von Menge der Objekte mit Sensorblickrichtung schräg nach hinten**

## 6 Zusammenfassung und Ausblick

Die Bestimmung der Fahrzeuggeschwindigkeit ist nicht nur im Automobile-Bereich sondern auch im Schienenverkehr-Bereich ein wichtiges Thema. Moderne Automotive-Radarsensoren stellen aufgrund ihrer geringen Größe und des Preises sowie der Möglichkeit zur Installation auf dem Wagenkasten eine Möglichkeit zur vorübergehenden Ausrüstung von Schienenfahrzeugen für Messkampagnen dar. Nachteil der Sensoren ist, dass diese üblicherweise in Fahrzeugmitte mit freier Sicht in Fahrtrichtung montiert werden.

Im Rahmen dieser Arbeit sollte die Eignung von Automotive-Mid-Range-Radarsensor für den Einsatz auf Schienenfahrzeugen untersucht werden, mit Sensorblickrichtung sowohl in der Fahrzeugfahrtrichtung als auch schräg nach vorne und hinten. Zuerst wurden die Eigenschaften bzw. die Ausgangsparameter des Sensors analysiert. Danach wurde ein Algorithmus erdacht um die Fahrzeuggeschwindigkeit zu berechnen. Nach der Vorbereitung und Durchführung von Messfahrten, wurden die verarbeiteten Messergebnisse mit anderen vorhandenen Sensoren verglichen um die Eignung bzw. die Messqualität zu überprüfen.

Anhand der vorliegenden Arbeit wird gezeigt, dass die Verwendung von Automotive-Mid-Range-Radarsensoren auf Basis von Messungen der radialen Relativgeschwindigkeit von umliegenden Objekten eine Möglichkeit bietet, die Fahrzeuggeschwindigkeit von Schienenfahrzeugen zu bestimmen, wenn die Sensorblickrichtung in Fahrzeugfahrtrichtung zeigt. Wenn die Sensorblickrichtung schräg nach vorne oder hinten zeigt, kann der Sensor wegen des Mangels an benutzbaren Objektgeschwindigkeiten nicht für die Bestimmung der Fahrzeuggeschwindigkeit eingesetzt werden.

Laut der Anweisung für den Eisenbahnbetriebsdienst des Testgeländes wurde das Fahrverhalten des Messfahrzeuges ähnlich wie im Rangierbetrieb mit einer maximalen Messgeschwindigkeit von ca. 30 km/h durchgeführt. Als Ausblick auf die Fortsetzung der Arbeiten kann die Messqualität für noch schnellere Fahrten in einem anderen Schienenfahrzeugtestfeld weiter untersucht werden.

## Literaturverzeichnis

- [1] GRIMM, Matthias ; HARTWIG, Katrin ; MEYER ZU HÖRSTE, Michael: *Anforderungen an eine sicherheitsrelevante Ortung im Schienenverkehr*. In: 20. Verkehrswissenschaftliche Tage (2005)
- [2] MEYER ZU HÖRSTE, Michael ; GERLACH, Katrin: *Precision and Availability Evaluation for Road and Rail Localisation Platforms*. Praha : 2008
- [3] DLR News-Archiv 2008. URL [http://www.dlr.de/DesktopDefault.aspx/tabid-3432/7418\\_read-12874/7418\\_page-5](http://www.dlr.de/DesktopDefault.aspx/tabid-3432/7418_read-12874/7418_page-5) – Überprüfungsdatum 2018-02-21
- [4] KLUGE, A. ; LÜDDECKE, K.: *Mobiles Labor RailDrIVE – synchrone Erfassung von Sensordaten*. EI – DER EISENBAHNINGENIEUR (Hrsg.), 001/2014
- [5] WOLFF, Marcus: *Sensor-Technologien : Band2 Geschwindigkeit, Durchfluss, Strömungsfeld*. Berlin, Boston : De Gruyter Oldenbourg, 2018
- [6] MANUEL SCHWABE: *Auffinden und Verfolgen von Verkehrsteilnehmern für autonome Fahrzeuge mittels Radar*. Berlin, Freie Universität Berlin. Diplomarbeit. 2012
- [7] TIPLER, Paul Allen ; MOSCA, Gene: *Physik für Wissenschaftler und Ingenieure*. 2., dt. Aufl., rev. Nachdr. Berlin : Spektrum Akad. Verl., 2007
- [8] SABINE REINHARDT: *Der Doppler-Effekt : Übungen im Vortragen mit Demonstrationen - Physikalische Chemie*. Bayreuth, 04.10.2013. URL <http://daten.didaktikchemie.uni-bayreuth.de/umat/dopplereffekt/dopplereffekt.htm> – Überprüfungsdatum 2018-02-25
- [9] FLORIAN FÖLSTER: *Erfassung ausgedehnter Objekte durch ein Erfassung ausgedehnter Objekte durch ein Automobil-Radar*. Hamburg, Technischen Universität Hamburg-Harburg. Dissertation. 2006
- [10] HERING, Ekbert ; MARTIN, Rolf ; STOHRER, Martin: *Physik für Ingenieure*. 6. Aufl. Berlin : Springer, 1997
- [11] Norm ISO 8855. Januar 1994. *Fahrzeugdynamik und Fahrverhalten Begriffe DIN 70000*
- [12] ROBERT BOSCH GMBH: *Produktdatenblatt Mittelbereichsradaresensor (MRR)*. URL <http://www.bosch-mobility-solutions.de/de/produkte-und-services/pkw-und-leichte->

- nutzfahrzeuge/fahrerassistenzsysteme/vorausschauendes-notbremssystem/mittelbereichsradsensor-(mrr)/ – Überprüfungsdatum 2018-02-25
- [13] SMG TECHNOLOGIE GMBH: *Betriebsanleitung : optische Weg- und Geschwindigkeits-sensoren*. LUXACT Neo 1D. 2014
- [14] IMC MEßSYSTEME GMBH: *LUXACT : Optischer Sensor zur berührungslosen Weg- und Geschwindigkeitsmessung*. URL [https://www.imc-berlin.de/fileadmin/Public/Products/Brochures/de/LUXACT\\_Broschuere\\_09\\_02.pdf](https://www.imc-berlin.de/fileadmin/Public/Products/Brochures/de/LUXACT_Broschuere_09_02.pdf) – Überprüfungsdatum 2018-03-08
- [15] BAUMER GMBH: *Lagerloser Achsimpulsgeber*. URL [https://www.baumer.com/de/de/lagerloser-achsimpulsgeber-fur-verbesserte-traktion-bei-schienenfahrzeugen/n/news\\_bearingless-axle-encoder](https://www.baumer.com/de/de/lagerloser-achsimpulsgeber-fur-verbesserte-traktion-bei-schienenfahrzeugen/n/news_bearingless-axle-encoder) – Überprüfungsdatum 2018-02-25
- [16] WIKIPEDIA: *Dilution of precision*. URL [https://de.wikipedia.org/wiki/Dilution\\_of\\_Precision](https://de.wikipedia.org/wiki/Dilution_of_Precision) – Überprüfungsdatum 2018-03-10
- [17] IAN POOLE: *GPS Accuracy, Errors & Precision*. URL <http://www.radio-electronics.com/info/satellite/gps/accuracy-errors-precision.php> – Überprüfungsdatum 2018-03-10
- [18] JAVAD GNSS INC.: *SIGMA-G3T Datasheet*. URL [https://www.javad.com/downloads/javadgnss/sheets/Sigma-G3T\\_Datasheet.pdf](https://www.javad.com/downloads/javadgnss/sheets/Sigma-G3T_Datasheet.pdf) – Überprüfungsdatum 2018-03-10
- [19] K. ETSCHBERGER: *Controller Area Network*. 3. Auflage. München : Hanser Verlag, 2006
- [20] LAWRENZ, Wolfhard ; OBERMÖLLER, Nils: *CAN Controller Area Network : Grundlagen, Design, Anwendungen, Testtechnik*. 5., neu bearbeitete Auflage. Berlin, Offenbach : VDE Verlag, 2011
- [21] ZIMMERMANN, Werner ; SCHMIDGALL, Ralf: *Bussysteme in der Fahrzeugtechnik : Protokolle, Standards und Softwarearchitektur*. 5., aktual. und erw. Aufl. Wiesbaden : Springer Vieweg, 2014 (ATZ / MTZ-Fachbuch)
- [22] MARC KLEINE-BUDDE: *SocketCAN - The official CAN API of the Linux kernel*. In: *CAN in Automation* (2012), S. 17–22

- [23] THE LINUX KERNEL ORGANIZATION: *Linux Kernel Documentation : can.txt*. URL <https://www.kernel.org/doc/Documentation/networking/can.txt> – Überprüfungsdatum 2018-02-25
- [24] CANUSB : *Universeller USB-CAN Interfaceadapter*. URL <https://elmicro.com/de/canusb.html>. – Aktualisierungsdatum: 2018-01-16 – Überprüfungsdatum 2018-02-26
- [25] LAWICEL AB, Sweden: *CANUSB Tools : CANUSB Download*. URL [http://www.can232.com/?page\\_id=75](http://www.can232.com/?page_id=75) – Überprüfungsdatum 2018-03-08
- [26] FUTURE TECHNOLOGY DEVICES INTERNATIONAL LIMITED: *Application Note : FTDI Drivers Installation Guide for Linux*. URL [http://www.ftdichip.com/Support/Documents/AppNotes/AN\\_220\\_FTDI\\_Drivers\\_Installation\\_Guide\\_for\\_Linux.pdf](http://www.ftdichip.com/Support/Documents/AppNotes/AN_220_FTDI_Drivers_Installation_Guide_for_Linux.pdf) – Überprüfungsdatum 2018-03-08
- [27] PASCAL WALTER: *Step-by-step guide: Installing the Lawicel CANUSB adapter on Linux*. URL <http://pascal-walter.blogspot.de/2015/08/installing-lawicel-canusb-on-linux.html> – Überprüfungsdatum 2018-02-26
- [28] LAWICEL AB: *CANUSB - Manual : Version 1.0D*. November 2011
- [29] DANNY COHEN: *Internet Engineering Note 137 : ON HOLY WARS AND A PLEA FOR PEACE*. URL <https://www.ietf.org/rfc/ien/ien137.txt> – Überprüfungsdatum 2018-02-27
- [30] RACELOGIC: *Vehicle CAN Bus Interface RLVBCAN01 : Instruction Manual*. URL [http://www.racelogic.co.uk/\\_downloads/vbox/Manuals/Input\\_Modules/RLVBCAN01\\_Manual.pdf](http://www.racelogic.co.uk/_downloads/vbox/Manuals/Input_Modules/RLVBCAN01_Manual.pdf) – Überprüfungsdatum 2018-02-26
- [31] R. VAN UFFELEN: *Technical Customer Documentation : MRRevo14F*. Bosch Engineering, 2015
- [32] FIKRET SENTÜRK: *Durchführen von Fahrversuchen hinsichtlich einer Optimierung von FHTW-Fahrdynamikfahrzeug*. Berlin, Fachhochschule für Technik und Wirtschaft Berlin. 2005
- [33] IHME, Joachim: *Schienenfahrzeugtechnik*. Wiesbaden : Springer Vieweg, 2016
- [34] WOLFRAM BÄUMER: *Anweisung für den Eisenbahnbetriebsdienst für die Hafenbahn Braunschweig*. URL [http://www.braunschweig-hafen.de/downloads/2013-10-14\\_HBG\\_BS\\_Bedienanw.pdf](http://www.braunschweig-hafen.de/downloads/2013-10-14_HBG_BS_Bedienanw.pdf) – Überprüfungsdatum 2018-02-28

- [35] GROOS, Jörn Christoffer ; ROTH, Michael Helmut ; HAVRILA, Patrik: *Zustandsüberwachung mit kostengünstigen Multi-Sensor-Systemen*. In: *EI - Der Eisenbahningenieur* (2017), S. 41–45
- [36] DAMIEN LHOMME-DESAGES, CHRISTOPHE GRAND, FAIZ BEN AMAR AND JEAN-CLAUDE GUINOT: *Doppler-based Ground Speed Sensor Fusion and Slip Control for a Wheeled Rover*. In: *IEEE/ASME TRANSACTIONS ON MECHATRONICS* August 2009, VOL. 14, No. 4

## Anhang

### Anhang A: Lageskizze des Testgeländes

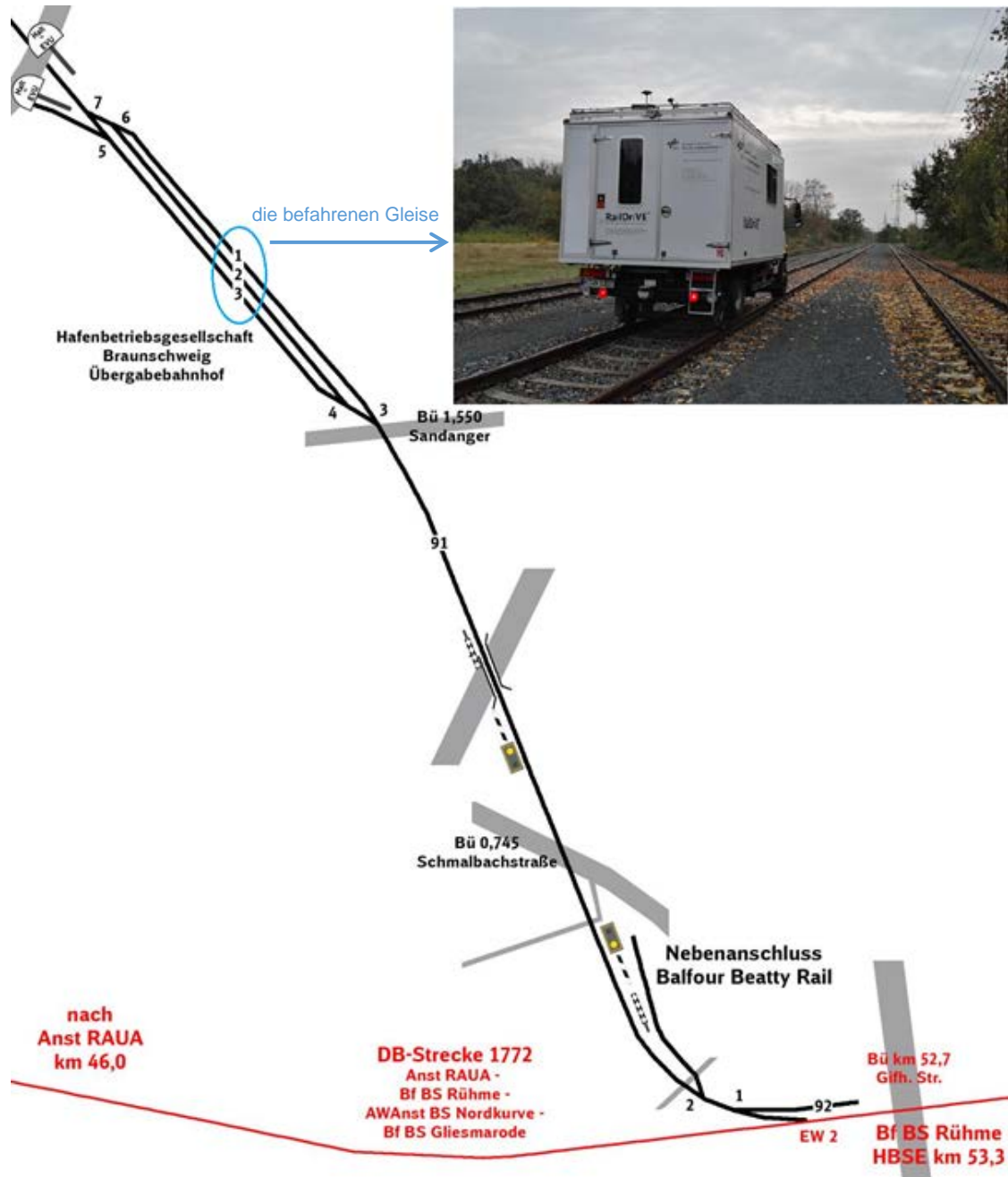


Abbildung A-1: Lageskizze des Testgeländes (vgl. [34])

## Anhang B: Quellcode des Algorithmus zur Bestimmung der Fahrzeuggeschwindigkeit

### B-1. Arithmetisches Mittel als Fahrzeuggeschwindigkeit

```

/*
arithmetisches Mittel der benutzbaren umgerechneten Relativgeschwindigkeiten
Vx_obj als Fahrzeuggeschwindigkeit anwenden
Wegen der Geheimhaltungsvereinbarung werden die Dekodierungsprozesse fortgelassen.
*/

// creat a function to get the absolute value of a number
float abs_value(float a) {
    if(a<0)
        return -a;
    else
        return a;
}

#define PI 3.14159

// radial velocity with respect to radar; oncoming object with negative sign; unit [m/s]
int object_vr = 0;
// radial distance with respect to radar; unit [m]
unsigned int object_dr = 0;
// azimuth angle, measured from the center of the sensor, left side is positive; unit [rad]
int object_phi = 0;

// indicate the begin of a can message cycle
if (can_id_hex == 0x3f2){
    // to store the calculated vehicle velocity in each cycle
    vr_cycle = 0;
    // to store the sum of usable velocity in each cycle
    vr_sum = 0;
    // to store the number of usable velocity in each cycle
    vr_counter = 0;
}
// ID of all the objects
// faktor_vr, faktor_dr, faktor_phi should refer to datasheet
if (can_id_hex > 0x3f2 && can_id_hex < 0x432) {
    //vr_value formula depends on the Installation Azimuth of the sensor
    //falls Sensorblickrichtung in Fahrtrichtung, Azi_sensor = 0 rad
    //falls Sensorblickrichtung schräg nach vorne, Azi_sensor = PI/4 rad
    //falls Sensorblickrichtung schräg nach hinten, Azi_sensor = 3 * PI/4 rad
    vr_value = faktor_vr * object_vr / cos( Azi_sensor + object_phi * faktor_phi);
    dr_value = faktor_dr * object_dr;

    // difference between current velocity and the previous velocity
    // useable can_msg filter, 0.174 m/s calibrated for RailDrive
    // "dr_value == 0" means no object for the certain CAN-ID
    if(dr_value != 0 && (abs_value(vr_value - vr_previous) < 0.174)) {
        vr_sum += vr_value;
        ts_cycle = ts; // timestamp of the object
        vr_counter++;
    }
}

```



```

// indicate the end of one can_msg cycle
// calculate the vehicle velocity
if(can_id_hex == 0x443 && vr_counter != 0) {
    // regard the mean value of the useable velocities as the velocity in this cycle
    vr_cycle = vr_sum / vr_counter;
    vr_previous = vr_cycle; // update the value of vr_previous
    cout<< ts_cycle <<" " << vr_cycle <<endl; // timestamp unit [ms], velocity unit [m/s]
}

```

## B-2. Median als Fahrzeuggeschwindigkeit

```

/*
Alternative: Median der benutzbaren umgerechneten Relativgeschwindigkeiten Vx_obj
als Fahrzeuggeschwindigkeit anwenden
*/

// generate a vector to store the usable vr_value in each cycle
vector<float> vr_value_v;

if (can_id_hex > 0x3f2 && can_id_hex < 0x432) {
    if(dr_value != 0 && (abs_value(vr_value - vr_previous) < 0.174) ) {
        // push the usable vr_value into vector
        vr_value_v.push_back(vr_value);
        ts_cycle = ts;
        vr_counter++;
    }
}

if(can_id_hex == 0x443 && vr_counter != 0) {
    // sort the elements in the vector
    sort(vr_value_v.begin(),vr_value_v.end());

    // find out the median value of the useable velocities as the velocity in this cycle
    if(vr_value_v.size() % 2 == 0){
        vr_cycle = ( vr_value_v[(vr_value_v.size()/2 -1 )] +
vr_value_v[(vr_value_v.size()/2)] )/2;
    }
    else{
        vr_cycle = vr_value_v[(vr_value_v.size()/2)];
    }

    vr_previous = vr_cycle;
    cout<< ts_cycle <<" " << vr_cycle <<endl;
    vr_value_v.clear(); // clear the vector at the end of each cycle
}

```